Multi -Threaded Web Crawler based on Multi keyword

Web Crawling using Ontology

Sachin Chirgaiya & Jayendra Barua
Department of Computer Science And Engineering
Malwa Institute of Technology, Indore, India
E-mail : sachin.chirgaiya@gmail.com, jayendrabarua@gmail.com

Abstract – Web crawlers facilitate the search engine's work by following the hyperlinks in Web pages to automatically download a partial snapshot of the Web. Crawling is the initial and also the most important step during the Web searching procedure. A prototype framework of a multi-threading Web crawler is proposed. Also the design and implementation of a multi-threading Web crawler is described and discussed. The end result of crawling is a collection of web pages at a central location[22].The experiment result demonstrates this prototype of Web crawler has better performance. It is possible to measure the performance of a search by understand user interest and information relevant The vast collection of computer networks which form and act as a single huge network for transport of data and messages across distances over the internet. The World Wide Web is an interlinked collection of billions of documents formatted using HTML. The user has to shift through scores of pages to come upon the information he/she desires[18]. Web crawlers are the heart of search engines. Web crawlers continuously keep track on crawling the web and find any new web pages that have been added to the web, pages that have been removed from the web.

Keywords – Search engine, Crawler, Semantic web, Ontology, Machine Learning

I. INTRODUCTION

Web crawling is the process by which we gather pages from the Web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. A Web crawler is one type of bot, or software agent. In general, it starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam). Different types of crawlers and the different techniques used make one to consider different issues while designing and implementing them.

Search engines use crawlers to find what is on the Web and then they construct an index of the pages that were found. Surveys indicate that almost 25% of Web searchers are unable to find useful results in the first set of URLs that are returned. The term ontology [1] is an old term used in the field of Knowledge Representation, Information Modeling, etc. Typically ontology is a hierarchical data structure containing relevant entities, relationships and rules within a specific domain. Tom R. Gruber [2] defines ontology as a specification of a conceptualization.

A. Search Engine

A search engine creation is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been conducted on them. Furthermore, due to rapid
advance in technology and web proliferation, creating a web search engine today is very different from past [3].

B. Crawler Based Search Engines

Crawler based search engines create their listings automatically. Computer programs ‘spiders’ build them not by human selection. They are not organized by subject categories; a computer algorithm ranks all pages. Such kinds of search engines are huge and often retrieve a lot of information -- for complex searches it allows to search within the results of a previous search and enables you to refine search results. These types of search engines contain full text of the web pages they link to. So one can find pages by matching words in the pages one wants.

C. Web Crawling

The Web crawler is started with a given set of links. The links are retrieved in the order of their rank. The Web documents are then passed on to the preprocessing module. To ensure a smooth retrieval a buffer of links, several parallel retrieval threads and a Web document buffer are utilized. Links and documents have to be checked to prevent doubles due to redirected links or mirror sites [4]. In most popular search engines, it is well-known that the ranks of websites for general queries are directly relevant to their economic benefits. Thus, a common phenomenon has emerged that many spam websites play tricks on search engine crawlers by artificially increasing links, in order to increase their weight in the Page Ranks algorithm [10]. Semantic web technologies concentrate on analyzing the semantics of web document content, which could be helpful to solve this issue. [11].

II. RELATED WORK

A web search engine is a tool designed to search for information on the World Wide Web. Its research could be derived from 1990s. One of the first "full text" crawler-based search engines was WebCrawler, which came out in 1994. Unlike its predecessors, it let users search for any word in any webpage, which has become the standard for all major search engines since. Several companies entered the market spectacularly, receiving record gains during their initial public offerings. Some have taken down their public search engine, and are marketing enterprise-only editions, such as Northern Light. Many search engine companies were caught up in the dot-com bubble, a speculation-driven market boom that peaked in 1999 and ended in 2001. Nowadays, a series of search engines are booming out, such as Yahoo, Google, Live search and so on.

Data mining will not uncover patterns that are present in the domain, but not in the sample. Humans have been "manually" extracting information from data for centuries, but the increasing volume of data in modern times has called for more automatic approaches. As data sets and the information extracted from them has grown in size and complexity, direct hands-on data analysis has increasingly been supplemented and augmented with indirect, automatic data processing using more complex and sophisticated tools, methods and models. The proliferation, ubiquity and increasing power of computer technology has aided data collection, processing, management and storage. However, the captured data needs to be converted into information and knowledge to become useful. [17].

A. Simple Focused Crawler

Fig. 1: Methodology of simple focus crawler.

B. General Architecture

Roughly, a crawler starts with the URL for an initial page P0. It retrieves P0, extracts any URLs in it, and adds them to a queue of URLs to be scanned. Then the crawler gets URLs from the queue (in some order), and repeats the process. Every page that is scanned is given to a crawler that saves the pages, creates an index for the pages, or summarizes or analyzes the content of the Pages [19] [20] [21].

A selection policy: This states which pages to download as a crawler always downloads just a fraction of the Web pages; it is highly desirable that the downloaded fraction contains the most relevant pages and not just a random sample of the Web.

A re-visit policy: This state’s when to check for changes to the pages. The Web has a very dynamic nature, and crawling a fraction of the Web can take a really long time, usually measured in weeks or months. By the time a Web crawler has finished its Crawl, many
events could have happened. These events can include creations, updates, and deletions. From the search engine’s point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource.

A politeness policy: This state’s how to avoid overloading Web sites. Needless to say, if a single crawler were performing multiple requests per second and/or downloading large files, a server would have a hard time keeping up with requests from multiple crawlers.

A parallelization policy: This state’s how to coordinate distributed Web crawlers. A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as two different crawling processes can find the same URL.

C. Ontology based focused crawlers

Generally speaking, ontology-based focused crawlers are a series of crawlers which utilize ontologies to link the fetched web documents with the ontological concepts (topics), with the purpose of organizing and categorizing web documents, or filtering irrelevant webpages with regards to the topics. There have been several studies on ontology based focused crawlers, which are briefly described below: Ehrig and Maedche proposed an ontology-focused crawler [12] [13]. Two cycles are involved in the crawling framework. In the first cycle, users can define a crawling target by instantiating a domain specific ontology, and limit the crawling scope by providing the URLs of crawled websites. Based on the ontology and crawling scope, the focused crawler starts to work on retrieving data from those websites, and computing the relevance between the ontological concepts and the crawled data by means of TF-IDF algorithm. Its implementation – CATYRPEL is built upon KAON – a framework for ontology-based application development. Ardö introduced a focused crawler working for the ALVIS – an open-source prototype of peer-to-peer semantic search engine [14]. The focused crawler is used to retrieve, cluster and store relevant web pages by linking them to topics. Each topic is defined by ontology of terms. Relevance values between the terms and document texts are computed from both global (the whole database) and local (the topic ontology) perspective, by means of the topical PageRanks algorithm [15].

In addition, the webpage relations can be viewed by linking the webpages to the ontology concepts that appear in the webpages. THESUS aims to organize online documents by linking their URLs to hierarchical ontology concepts, which are seen as thematic subsets. A web crawler is used in the document acquisition component of the system. The mechanism of this crawler is as follows: first, the crawler extracts the URLs and their descriptive texts from the initial set of documents; then the descriptive text of one URL are matched with one of the ontological concepts, and the URL is linked to concept. A threshold of maximum times of recursions or maximum number of documents is set as an ending requirement[15]. As many ontology-based crawlers’ stored ontologies cannot completely define crawling targets, Su et al. designed an ontology-learning focused crawler. The ontology learning mechanism originates from the theory of reinforcement learning – a decision-making framework based on reward or punishment points. First of all, a weight namely the distance between a concept and a topic is predefined. Then the crawler starts to retrieve documents based on the relevance values between each document and the ontological concept, which is computed by considering both the concept’s weight and term frequencies in each document. An interest ratio to this topic is obtained by predicting the probability of a crawling event hitting the topic after the crawling process. Finally the weight is recomputed by considering an evolved weight based on the original weight and the interest ratio. From the above introductions, we can observe that most of these crawlers utilize various ontology document link analysis technologies to control crawling scope, cluster and retrieve web documents according to users’ specific interest[11].

III. PROPOSED WORK

A. Semantic Web

The next generation intelligent web called the semantic web offers users the ability to work on shared meaningful knowledge representations on the web. Semantic Web creates an artificial intelligence (AI) application which will make web content meaningful to computers, thereby unleashing a revolution of new abilities and it intends to support machine-processing capabilities that will automate web applications and services. Agents (software programs) will perform various tasks by communication with other agents and seeking information from web resources[6]. Sir Tim Berner’s Lee, dreams about human-machine human collaboration where machines would also be able to understand and analyze data and transactions on the web. It is a concept of how computers, people, and the
web can work together more effectively than is possible now[7].

B. Ontology

To compare conceptual information across two knowledge bases on web, a program must have a way to discover common meanings and the solution to this is to collect information at a common place called Ontologies. Ontology formally describes a list of terms which represent important concepts, such as classes of objects and the relationships between them in order to represent an area of knowledge[6]. Ontologies provide a formal semantics that can be employed to process and integrate information on the web. Gruber [7] describes ontology as an explicit specification of conceptualization[8]. Ontologies play a pivotal role in providing a vocabulary comprising unambiguous definitions for terms that can essentially serve as a formal support for communication between software agents. They provide a communication mechanism for users and software agents, clearly define the semantics for different domains for the purpose of interactions on the web, and help in creating a knowledge base that will enable people to work on a particular domain [9].Web Ontology Language(OWL), recommendation from W3C, is widely used to construct a domain ontology[5].

C. Machine Learning and Ontology

Ontology matching is the problem of finding the semantic mappings between two given ontologies. This problem lies at the heart of numerous information processing applications. Virtually any application that involves multiple ontologies must establish semantic mappings among them, to ensure interoperability. Examples of such applications arise in myriad domains, including e-commerce, knowledge management, e-learning, information extraction, bio-informatics, web services. Despite its pervasiveness, today ontology matching is still largely conducted by hand, in a labor-intensive and error-prone process. The manual matching has now become a key bottleneck in building large-scale information management systems. The advent of technologies such as the WWW, XML, and the emerging Semantic Web will further fuel information sharing applications and exacerbate the problem. Hence, the development of tools to assist in the ontology matching process has become crucial for the success of a wide variety of information management applications[16].

IV. METHODOLOGY OF FOCUSING A WEB CRAWLER WITH ONTOLOGY

In a hybrid approach for searching the Semantic Web is described: classical search techniques are combined with spread activation techniques applied to a semantic model of a given domain. Figure 3 is a high level representation of the context and applications of our crawler system. We can distinguish four important parts: (i) the input of the crawler, (ii) the ontology commitment by the user used to guide the crawler, (iii) the processing and storing of the processed input and (iv) a selection of the possible text of that processed data. It is important that all this semantic content can be shared and reused by humans and by artificial agents. Although currently only a small portion of the Web is annotated with metadata, over time, we expect a much larger percentage of the web will be annotated.
A. Crawling steps

Step 1: Get the seed URL from initURL.txt file and class keyword from OWL file.

Step 2: If the webpage is valid that is it of the defined type like (html, php, jsp etc.) then it is added to queue.

Step 3: Parse the content with the help of Jena Class Extractor with the help of html parser for all the keyword found from the Ontology classes.

Step 4: Get the response from server if it is ok add the webpage to index and caches file to a folder. With the help of cache and index searching can be done and ranking the page according to the keyword search.

Step 5: After crawling the current page extract the new URL’s from the page and store the URL in an ontology based database with the page rank.

B. Proposed Algorithm

1: Retrieve all classes of a keyword using ontology.

2: Maintain a variable AverageRank (initially = 1).

3: pick up a URL from URL_Buffer and retrieve corresponding web page.

4: Extract all URL(s) (Extracted_URLs) from webpage

5: Calculate the rank for each class(found using ontology in step 1).

6: Calculate TotalRank = Sum(rank of all classes).

7: if(AverageRank <= TotalRank) then add Extracted_URLs to URL_Buffer.

else (Discard all Extracted_URLs).

8: AverageRank = (AverageRank + TotalRank)/Total URL_Crawled.

9: Go to step (3)

C. Implementation

Initially, user inputs a seed URL to the web crawler. Crawler then fetches the corresponding web page from the Internet. This page is then stored in a buffer and its contents are parsed. We separate the links and content of the fetched web page. Now, the extracted links are stored in the database. Also, if we want to further traverse these links we can click the desired link and this link is given as input to the crawler and same procedure is repeated. Additionally, we can search a particular keyword in the content of the fetched urls stored in database and check if it is present or not. If yes, the crawler prints the number of times the particular keyword occurs in the Web Page.

V. RESULTS ANALYSIS AND CONCLUSION

In this section we have illustrated and briefly evaluated our relevance computation algorithm which is based over onto crawler concept. We have done this in a small and controlled environment so that the functioning of the algorithm could be clearly visualized and experienced. We perform a brief and empirical evaluation study of the crawler in an uncontrolled, practical environment, namely the Semantic Web. By performing such an uncontrolled empirical evaluation,
we want to point out and illustrate some interesting statistics concerning our crawler. We introduce the well-known harvest rate evaluation metric that is generally used to evaluate focused crawlers. The harvest rate can be denoted as \( h# \) which is the percentage of the web pages crawled satisfying the predicate. We present different scenarios to test the harvest rate of our crawler.

We compare the harvest rate on crawls using different kind of topic-ontologies. We also compare the harvest rate of a crawl using a topic-ontology (our approach) and a crawl focused around a simple keyword topic. In addition, we provide some statistics that give an idea of the performance that can be expected from the crawler. We also compare the computing time between all phases of the crawling loop. The introduced experimental data will give us an idea on the usefulness and applicability of the crawler on the current Semantic Web.

**VI. CONCLUSION AND FUTURE SCOPES**

In this paper, we briefly review the existing keyword Matching Crawler and Multi-Threaded Ontology-based focused crawlers. A comparison is made among these crawlers from the perspective of keyword matching and domain specific Crawling, working environment, special functions, technologies utilized, evaluation metrics and evaluation results, in order to survey its current research status. The conclusion with respect to this comparison is made in this section. For making Crawler more intelligent we can use the NLP which is a more effective concept for providing the real world meaning to the ontology based web crawler for the key word matching and searching that can be implemented in the further scope of this paper.

**VI. ACKNOWLEDGMENT**

I thank all the faculty members of my college for intensive discussions and contributions, especially from CSE Dept. In area of Artificial Intelligence and web mining.

**VII. REFERENCES**


