Design of a Migrating Crawler Based on a Novel URL Scheduling Mechanism using AHP

Deepika Punj, Computer Engineering Department, YMCA University of Science & Technology, Faridabad, India
Ashutosh Dixit, Computer Engineering Department, YMCA University of Science & Technology, Faridabad, India

ABSTRACT

In order to manage the vast information available on web, crawler plays a significant role. The working of crawler should be optimized to get maximum and unique information from the World Wide Web. In this paper, architecture of migrating crawler is proposed which is based on URL ordering, URL scheduling and document redundancy elimination mechanism. The proposed ordering technique is based on URL structure, which plays a crucial role in utilizing the web efficiently. Scheduling ensures that URLs should go to optimum agent for downloading. To ensure this, characteristics of both agents and URLs are taken into consideration for scheduling. Duplicate documents are also removed to make the database unique. To reduce matching time, document matching is made on the basis of their Meta information only. The agents of proposed migrating crawler work more efficiently than traditional single crawler by providing ordering and scheduling of URLs.

KEYWORDS
Duplicate, Hashing, Migrating Crawler, Ordering, Scheduling URL

INTRODUCTION

With the increase in size of web, it is necessary to ensure the richness and uniqueness of information available on it. In the era of ICT, search engine plays a vital role for finding relevant information (Sharma & Sharma, 2011). For relevant information both the parameters namely Users’ behaviour (Deepika and Dixit, 2015c) and rich database are significant. The focus of the present paper is on maintaining the database richer. The crawler is one of the main modules of search engine and it is responsible for gathering the information from web and stored in database. There are many design issues while designing the crawler like uniqueness & richness of database, cooperation between crawling agents, fast & efficient crawling etc (Deepika and Dixit, 2012). There are many types of crawler exist such as Parallel Crawler, Migrating Crawler, Focussed Crawler, Hidden Crawler, Incremental Crawler etc. In this proposed work, the capabilities of migrating crawler (Singhal et al., 2012) are being utilized for designing an efficient crawling system. The proposed Crawler tries to achieve all the design issues needed for efficient crawling. Here, an URL plays an important role in designing a migrating crawler. In migrating crawler environment, if an URL is scheduled properly it will improve the richness of database with least communication overhead on to the network.

The URL is composed of five components namely the scheme, authority, path, query and fragment components (Lee et al., 2005) as shown below:

DOI: 10.4018/IJRSDA.2017010106

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
There are URLs which are points to same page. This problem is designated as DUST (Schonfeld et al., 2007) i.e. different URLs with similar text. DUST effect the whole working of Search Engines i.e. crawling, indexing, ranking etc. In order to remove this duplicity at URLs level proper processing has been adopted. Before checking for duplicity in URLs, URL standard Normalization process (Lee et al., 2005) is applied to them. This process eliminates the syntactically similar URLs. There are three types of normalization process:

1. Case normalization
   Conversion of scheme component letters and hostname to lowercase is done in this type of normalization

2. Percent-encoding normalization
   All unreserved characters like ~, _ etc are decoded into %form.

3. Path segment normalization
   Remove all ‘.’, ‘..’ from the path component of the URL.
   Remove the fragment component from the URL i.e. after #.
   Eliminate port number like 80.
   Remove ‘/’ from the end and add ‘/’ at path location if it is null.
   With the help of these normalization processes, syntactically similar URLs are identified.

Sitemap (Deepika and Dixit, 2015a) is also used here to get the information of all links present in a webpage. Basically, sitemap gives the number of links present in a web page. This helps in ordering the URLs before downloading them. To maintain the database unique and rich, hashing algorithm is used. Hashing helps in eliminating duplicate documents from the database.

RELATED WORK

Some of the prevalent work in the related area is discussed below. Agarwal et al., (2009) worked on crawl logs and then from these logs generate rules. These rules were then utilized for finding duplicate web pages. They form clusters of similar pages from crawl logs. These clusters were then used to generate rules for detecting duplicate pages. These rules were then generalized and with the help of these rules then only URL were able to detect identical pages. They also showed that their proposal effects crawling, indexing in effective way.

Qureshi and Khan, (2013) observed some limitation in previous work. According to them, by taking signature of whole body text is time consuming. So, they considered to reduce the body text and then applied MD5 fingerprinting mechanism. They suggested some ways to reduce the body text and by doing this they reduced the time wastage in fingerprinting method.

Luis et al., (2013) proposed an algorithm for complex hierarchal datasets like XML datasets. He uses a Bayesian network to check whether the two XML documents are duplicates or not. It shows the probability of how many the documents are duplicates. The proposed technique not only works on content within the documents but also the structure of documents.

Wei Li et al., (2005) works on both the syntax and semantic part of web documents to eliminate duplicate web documents. It considers keyword sequences for syntax i.e. structure and semantics i.e. intension feature of web documents. The keywords sequences of different documents are compared to identify duplicate documents. If the comparison found them similar, then documents are considered as same and one of them is eliminated.
A Comparative Analysis of a Novel Anomaly Detection Algorithm with Neural Networks
*International Journal of Rough Sets and Data Analysis* (pp. 1-16).