CS298 Progress Report

Extending Yioop! abilities to search the Invisible Web

Introduction

This project aims to add to the Yioop search engine the ability to crawl and index the Invisible Web. The Invisible Web refers to the information like database content, non-text files, password restricted sites, etc. on the Web. One source of dark content on the Web is URL shortening services. It is a source of dark content as in ranking results search engines often attribute the link to the URL short link service rather than to where the link points to. The goal is to study how Yioop deals with these links and add to Yioop the ability to associate these links with the original links. Sphinx is an open source search engine which has the ability to index database content. The aim is to study how Sphinx does this and extend Yioop to have the same ability to crawl the database content as Sphinx. The project also aims on adding to Yioop the ability to crawl password restricted sites provided the username and password are given. Also, I will work on adding extension to Yioop to extract URL links from JavaScript links on websites. Solving these problems and adding these extensions would extend the ability of Yioop to search into the Invisible Web.

Work done in Spring, 2011

In the beginning of the semester, I started working on my first deliverable of CS298 which involved extracting URLs from JavaScript. In order to accomplish this task, I had to extract the script tags and the content of the script tags from the source of a webpage. After completing this task, I realized that I will need a JavaScript interpreter which can execute the JavaScript content in the script tags and give the result which can then be used to extract the URLs. I have explored various JavaScript interpreters for this reason. I decided on using Spider Monkey for the same. I started implementation of this deliverable using Spider Monkey. I had faced a challenge during this process as Spider Monkey does not provide interpretation for document, window or navigator objects of JavaScript. In order to solve this problem, I started implementing my own functions in JavaScript for the necessary objects. This deliverable is in coding phase.

I have also started working on the second deliverable of crawling password restricted sites. In order to accomplish this task, I had to create a UI in order to provide the user a place to enter the URLs, username and passwords. I had created a UI for the same in the "Manage Crawls" section of Yioop and saved the usernames and passwords in the backend. Later, I had realized I could have the users enter the username and password in the seed sites section of Yioop using a specific syntax. I have to store these details and write code to implement the deliverable.

Conclusion and Roadmap for finishing the Project

The next task would be to implement the remaining code and start my third deliverable. I need to implement all the functions of the objects in the first deliverable and add code in the queue_server.php and fetch_url.php for the second deliverable.

The Spring semester was spent completely working on the first two deliverables and I have faced challenges working on the first deliverable. Due to few problems, I could not complete the project as planned and I would be completing it in the next semester.