University of Glasgow | School of Computing Science

Efficient Web Crawler for Forum Sites

Tan Ming Sheng

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — February 18, 2015

Abstract

Forums have become an important resource on the web due to its increasing richness of user generated information through the contribution by millions of internet users on a daily basis, therefore harvesting forum data can allow analyst to discover useful information to help improve business intelligence. In order to efficiently harvest customised data, one would require certain level of technical knowledge to understand the complex APIs and using them is also programming effort intensive. The aim of this project is to develop an interactive user centric crawler system to allow users to visually select and extract usable information (i.e. text, structured data, videos and images) in an intuitive manner from any forums or websites.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

1.1 Background

1.2 Aim

1.3 Outline

Chapter 2

Related Work

Research has been done on related works to identify the current trends and problems in the market.

2.1 Generic Crawler

2.1.1 HTTrack

HTTrack [3] is a website copier and it basically mirrors a whole website by crawling a single or a list of targeted URLs and it also crawls the links inside the targeted URLs which makes it good for offline browsing but it does not allow users to choose specifically what they want inside the targeted URLs, furthermore, it requires installation of the software which means it can only be used locally on the users machine and the crawling would stop if the machine is down.

2.2 Focus Crawler

2.2.1 FoCUS

Chapter 3

Requirements

3.1 Supervisor's Requirements

- Explore and develop an real-time automatic forum crawler for fetching data from 10 web forums from Singapore. - Pre-process the retrieved data and store it in a database. - A dashboard is also required to be developed to allow users to manage and monitor the prototype crawler system - Depending on progress of the project, student should also conduct research and explore the possibility of customizing intelligence into the crawler to enable it to understand the content and the structure of a forum site, and then decide how to choose traversal paths among different kinds of pages.

3.2 Proposed System

A distributed vertical crawler [4] uses a specific set of pre-defined templates for crawling different kinds of websites in a distributed environment however, there are too many websites with different kinds of template and the pre-defined templates may not necessarily be the most efficient way in extracting information of what the users want. It leads to the creation of a prototype, codename Focra (i.e. Forum crawler) which is a dynamic visual user-defined template crawler to demonstrate the key ideas of this project.

Chapter 4

Design

4.1 User Interface

Focra allow the user to preview their candidate data [1]

4.2 System Architecture

4.2.1 HTML-Preprocessing

4.2.2 User-Defined Template

4.2.3 Structured Information Extraction

4.3 Class Diagram

Chapter 5

Prototype Implementation

The prototype which uses Django (web framework), Scrapy (crawler) and MongoDB (database) is currently able to allow different users to remotely manage their crawlers by using their browsers and the server will process the crawling for the users. Focras development source code has been uploaded to GitHub at [5].

5.1 Visual Selection of Template

5.2 Pagination

5.3 Nested Crawling

5.4 State Saving

Chapter 6

Evaluation

6.1 Usability Test

6.2 Performance Test

Chapter 7

Conclusion

7.1 Summary

7.2 Future Enhancement

Appendices

Appendix A

Running Focra

An example of running from the command line is as follows:

```
> python manage.py runserver
```

This will apply $BBMC$ with $style = 1$ to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

Appendix B

User Guide

Bibliography

[1] Tsuyoshi S. and Yuzuru T. Interactive Web-Wrapper Construction for Extracting Relational Information from Web documents. In *Proceedings of the 14th international conference on World Wide Web*, pages 968–969, 2005.