Analysis of Kenyan Academic Websites using Webometric Algorithms

Eric Chomba Ng'ang'a
P15/1263/2010

School of Computing and Informatics

University of Nairobi

February 7, 2014

B Sc (Computer Science)

Submitted in partial fulfillment of BSc (Computer Science)

Declaration

The project presented in this report is my original work and has been done with the full support of my supervisor as part of the fulfillment for the Diploma of Science in Computer Science

Signed ..... Date .....

author

$regno

This Project has been submitted as part fulfillment for the Diploma of Science in Computer Science with the approval as a Lecturer at the University of Nairobi, School of Computing and Informatics and as the Project Supervisor.

Signed ..... Date .....

supervisor

Acknowledgement

I sincerely thank everyone who contributed to the success of this Project. I thank my supervisor author, for being available and resourceful. His positive contributions helped improve the project. I also thank my colleagues in general for the useful suggestions, constructive criticisms and overall support during the period I undertook the project. ——Mr. Tuka———- Thank you all.

Abstract

abstract goes here

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API Application Programming Interface
DOS Denial of Service
DDOS Distributed Denial of Service
URL Uniform Resource Locator

Chapter 1

Introduction

1.1 Background

The World Wide Web is an expansive system of interlinked information including text, audio and visual media. This information is stored in web servers and accessed using a web browsers provided there is a connection from the web browser to the web server e.g. internet connection. A website is a collection of related web pages served from a single web domain hosted by one or more web servers. The World Wide Web is composed of all websites that are available publicly.

In today's era of the internet, websites have become a major avenue through which information is shared. Websites range from fully fledged web applications to simple personal pages to complex institution pages. Companies, institutions and individuals use to websites to post information about themselves, topics of interest, ideas etc. Websites have become so integrated within the society that they are now being used to achieve more than just information sharing. Reasons for developing websites are as many as the uses of websites. They include communication, information sharing, business, banking, entertainment, gaming and many more.

Endeleza hii sentensi

More and more researchers and students use websites to find scholarly material.

Most of the information published in websites is targeted at a certain audience.

Examples of previous statement

For example, However, the information providers may not know if the information is effective in achieving their goal of information dispersing to their target audience. Thus, website owners and administrators need a tool to help them assess the impact of their website. This project aims to help website owners to analyse and determine the effectiveness of their websites.

1.2 Problem Statement

This project will focus on one institution, the University of Nairobi (UoN). The institution was chosen because there has been a need for analysis of it's website within the institution itself. The university's website is particularly important since it should reflect the academic excellence of the university.

The university has a system of websites under the domain 'uonbi.ac.ke'. Under the domain 'uonbi.ac.ke', units within the University of Nairobi have sub domains. These units include colleges, schools, faculties, institutes, centres, university staff and student organizations and university service providers. The system of websites is governed by a policy. According to the university's website policy, the main objectives for the website are:

- To ensure accuracy, consistency, integrity of the content and protection of the identity and image of the University.
- To improve the University's visibility regionally and internationally and create a strong brand in line with the University's Strategic Plan.
- To provide a set of mandatory guidelines for the University of Nairobi System of Websites
- To guide the maintenance of the web content and evolution of the System of Websites to ensure continued reflection of the true status of the University within its web space.

In order to know the effectiveness of the system of websites, analysis needs to be done to determine the impact of the system of websites. An example of this analysis is the webometrics ranking of universities in which.

Currently, there is no effective method to analyse and determine the effectiveness and impact of the system of websites by the university itself. This is especially important to learning institutions since they need to know the impact they have on their intended audience. Such analysis would be helpful in determining whether the system of websites is upholding the set policy.

1.3 Project Justification

Provision of information about the effectiveness and impact of websites will enable website administrators to make more informed decisions. For example, they would know when to improve on their website or policies governing the same.

Do it

Add webometrics objectives

Chapter 2

Literature Review

2.1 Background

Analysis of websites has been carried out in various methods. These forms can be broadly classified into two major categories: qualitative and quantitative methods. Qualitative methods are methods which mainly deal with gaining a deeper understanding about websites and interaction with humans, producing qualitative data. They include content analysis, focus groups, interviews, referrer analysis, user feedback and audience analysis. Quantitative methods employ statistical, mathematical or computational techniques to study websites, producing quantitative data. They include website log analysis, bibliometrics, scientometrics and webometrics.

Most web analyses are based on quantitative data. This quantitative data contributes to general knowledge about the usage of a given website. This data can be broken down into numbers and graphs to be interpreted by the website administrators.

Website log analysis is done on logged data e.g. in log files generated from traffic to a website. It is shows details like the number of visits, origin, distribution and their referrals. It is one of the most preferred methods when it comes to website analysis. Most of the current website log analyzers offer a variety of visualization channels to show relevant demographic and geographic data of website visitors. One of the most popular tools to use in this analysis is Google Analytics. This is a tool that allows website administrators to view statistics about their website based on logged data over a period of time. Website administrators use a special code in their web pages and applications to track visitors and generate statistics. Another tool available is AwStats. Unlike Google Analytics, AwStats uses server log files to retrieve statistics about traffic to the website.

However, website log analysis is not a very credible measure of website impact. For example, academic websites usually target a small group of people. The number of visits to pages within such a website may not truly reflect the impact on the target group.

Webometrics is another major website analysis method. According to Michael Thelwall, webometrics is “a set of quantitative techniques for tracking and evaluating the impact of web sites and online ideas”. Webometrics encompasses the following fields:

> Informetrics. This is "the study of the quantitative aspects of information in any form, not just records or bibliographies, and in any social group, not just scientists."
>
> Bibliometrics. This is "the study of the quantitative aspects of the production, dissemination and use of recorded information"
>
> Scientometrics. This is "the study of the quantitative aspects of science as a discipline or economic activity"
>
> Cybermetrics. This is "the study of the quantitative aspects of the construction and use of information resources, structures and technologies on the whole Internet, drawing on bibliometric and informetric approaches"

The terms ‘webometrics’ and ‘cybermetrics’ are often used synonymously though webometrics is a subset of cybermetrics.

The main method used in webometrics is link analysis. The reason is that the links to a web site can reveal useful information about how popular it is, which pages or resources are the most popular, why it is popular and where it is popular. Whilst all this information can also be gained from web server log file analysis, the latter can normally only be conducted with permission of a site’s webmaster. In contrast, link analysis can be applied to any web site. This means that link analysis can be used to evaluate a web site by comparing it to its competitors or to similar web sites and can also be used to identify missed audiences for a site. Link analysis has been used extensively to develop algorithms like the Page Rank algorithm, which is used to rank websites in Google’s search engine.

More info on link analysis

More info on link analysis

Link analysis on academic institution websites has been inspired by bibliometrics and citation analysis. A lot of information is published on an academic website. Although some academic papers are published on university web servers, there is a lot of other information in university web sites that is not part of the academic publishing process and is not attempting to directly contribute to the progress of academic knowledge. This will be common knowledge to anyone that has used a university web site, but is nevertheless important. Middleton, McConnell, and Davidson (1999) have proposed, "a model for the structure and content of a university [web] site", claiming that it is in the interests of a university to provide three different types of information.

> Promotional information: advertising services, assets and achievements to potential customers, collaborators and recruits (recruits being both staff and students).

Value-added information: providing genuinely useful services to people, encouraging their return and enhancing the institution's reputation as an innovative information provider

Utility: to staff and students: information, services and resources that will enable an institution to reach it strategic aims more easily, facilitate external and internal communication and enhance education. This may have the additional benefit of impressing potential customers and recruits, demonstrating the facilities which will be available to them, should they choose to come to the institution.

Self-promotion is an important part of the wider process of research, even for individual academics (Hyland, 2003). In other words, departments that do not publicize their work to a more general audience than those that read their published articles are not optimizing their research 'in the round', and may loose out on such things as contacts with industry and new students. The same study also claims that institutions should provide: ...space for scholarly use and learning new ways of exploiting the new medium. Everybody within the institution should have the ability/opportunity to feed into the web site, provided sufficient editorial guidelines are in place. This promotes a vibrant web culture that encourages usage - seen by many as the raison d'etre of the Internet. Again, this contribution is unlikely to be surprising for many readers, but it does emphasize that experimentation and variety are natural to university web sites.

2.2 Webometrics Ranking of World Universities

This is one of the most notable implementation of webometrics. It is an undertaking by the Cybermetrics Lab (Spanish National Research Council, CSIC). It analyses and ranks over 20,000 university or academic institution websites. The rankings are published twice a year (January and July) since 2004. The rankings are based on web impact, web presence, openness and excellence.

Web Impact

The quality of the contents is evaluated through a "virtual referendum", counting all the external inlinks that the University webdomain receives from third parties. Those links are recognizing the institutional prestige, the academic performance, the value of the information, and the usefulness of the services as introduced in the webpages according to the criteria of millions of web editors from all over the world. The link visibility data is collected from the two most important providers of this information: Majestic SEO and ahrefs . Both use their own crawlers, generating different databases that should be used jointly for filling gaps or correcting mistakes. The indicator is the product of square root of the number of backlinks and the number of domains originating those backlinks, so it is not only important the link popularity but even more the link diversity. The maximum of the normalized results is the impact indicator.

Web presence

The total number of webpages hosted in the main webdomain (including all the subdomains and directories) of the university as indexed by the largest commercial search engine (Google). It counts every webpage, including all the formats recognized individually by Google, both static and dynamic pages and other rich files. It is not possible to have a strong presence without the contribution of everybody in the organization as the top contenders are already able to publish millions of webpages. Having additional domains or alternative central ones for foreign languages or marketing purposes penalizes in this indicator and it is also very confusing for external users.

Openness

The global effort to set up institutional research repositories is explicitly recognized in this indicator that takes into account the number of rich files (pdf, doc, docx, ppt) published in dedicated websites according to the academic search engine Google Scholar. Both the total files Both the total records and those with correctly formed file names are considered (for example, the Adobe Acrobat files should end with the suffix .pdf). The objective is to consider recent publications that now are those published between 2008 and 2012 (new period).

Excellence

The academic papers published in high impact international journals are playing a very important role in the ranking of Universities. Using simply the total number of papers can be misleading, so we are restricting the indicator to only those excellent publications, i.e. the university scientific output being part of the 10% most cited papers in their respective scientific fields. Although this is a measure of high quality output of research institutions, the data provider Scimago group supplied non-zero values for more than 5200 universities (period 2003-2010). In future editions it is intended to match the counting periods between Scholar and Scimago sources.

2.3 Indicators

These four indicators give an academic institution's visibility and activity. The visibility is gotten from impact and accounts for 50%. The activity is comprised of web presence, openness and excellence and accounts for 50%. The Webometrics ranking methodology comprises of four different parameters; Visibility (50% (link visibility 20% and G-factor 30% as of July 2011)), Size (20%), Rich Files (15%), and Scholar (5%) and Scimago (10%).

pie chart img

get better pie chart

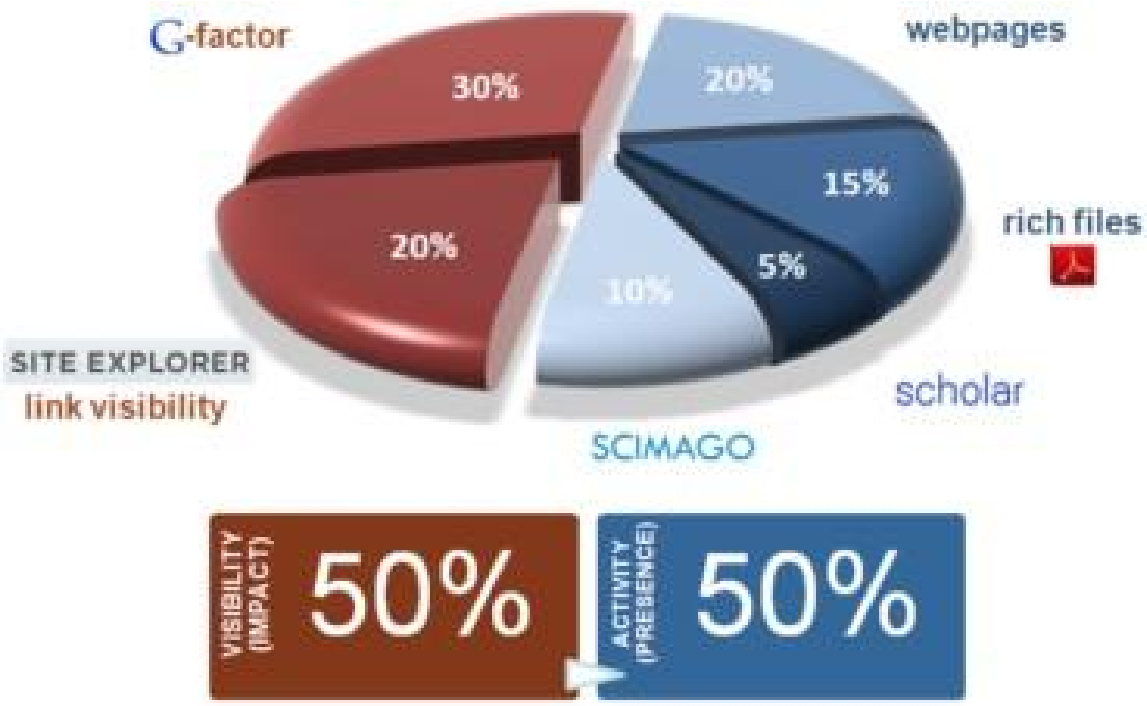

Figure 2.1: Webometrics university ranking methodology. Source :

Visibility

The total number of unique external links received (inlinks) by a site can be only confidently obtained from Yahoo Search. Results are log-normalised to 1 for the highest value and then combined to generate the rank.

Size

Number of pages recovered from four engines: Google, Yahoo, Live Search and Exalead. For each engine, results are log-normalised to 1 for the highest value. Then for each domain, maximum and minimum results are excluded and every institution is assigned a rank according to the combined sum.

Rich Files

After evaluation of their relevance to academic and publication activities and considering the volume of the different file formats, the following were selected: Adobe Acrobat (.pdf), Adobe PostScript (.ps), Microsoft Word (.doc) and Microsoft Powerpoint (.ppt). These data were extracted using Google and merging the results for each filetype after log-normalising in the same way as described before.

Data storage

Data collected from the websites crawled had to be stored for future analysis. Storage was possible in two ways :

1. **Traditional relational databases** They provide the most common solution for storing data. They aim to provide data integrity even though this is done while compromising performance. The web crawler collects numerous pages per second on a good network connection. Therefore, the crawler generates alot of data in a short amount of time. This would lead to the system hitting the database with inserts multiple times per unit time and the RDBMS trying to validate and ensure data integrity. This would undoubtedly develop into a bottleneck for the system in terms of scalability and speed.

2. **Flat files** They provide the simplest solution of the three choices. The format chosen to store the data was JSON. The data was stored as JSON lines not JSON objects. This was done to ensure the reading and writing of the files was done on a line by line basis not as one object to ensure minimal memory consumption and I/O access delays. The system at first used json files to store the data though it became evident that file location would become a thorny issue.

3. **NoSQL databases** They come in different forms e.g. document-oriented databases, graph databases e.t.c. They were developed as a solution to the constrictions of traditional SQL databases'.

comparisons

comparison between sql and nosql databases reasons for using flatfiles (json) comparison between document and graph databases

Chapter 3

Methodology

3.1 Research methodology

3.2 System methodology

Why evolutionary, advantages,disadvantages

Evolutionary prototype methodology was chosen to implement this project. The main goal in this methodology is to create a robust prototype in a structured way and constantly refine it according to the requirements which would evolve into the final system. The methodology was chosen because of its high tolerance to changing requirements. Only the requirements that are well understood are handled first. The methodology has four main stages:

1. Definition of basic requirements
2. Developing working prototype
3. Verifying prototype
4. Changing requirements

This methodology was especially helpful in overcoming a hiccup encountered by the developer when one of the requirements gathering tools was delayed which consequently delayed the collection of requirements. Some advantages of the methodology include:

- User involvement from the start
- Suitable for projects with vague and changing requirements
- User may start using system early in development stage

disadvantages

The main disadvantages of the methodology are

3.2.1 Process Overview

system development process overview

3.2.2 Iterations

During each iteration, the prototype will be checked against the set requirements. Changes will be made accordingly to fit the requirements. Functionality will be added incrementally to the prototype. Testing shall be done to the additional functionality before integrating with the prototype. After passing the tests, the new functionality will be integrated to the system. Testing shall be done on the prototype after integration with the new functionality in each iteration to ensure stability of the proposed system through the iterations.

system development iteration overview

The major iterations in development of the system were :

1. Developing web crawler module
   This is the first iteration of the system development process. The web crawler was developed so as to collect data from the web servers and search engines to be used in the analysis. The crawler was developed in two major stages:

   (a) Standalone web crawler. The crawler was designed and implemented as a single standalone crawler. it handled crawling of the given websites synchronously (one at a time) and eventually asynchronously (two websites concurrently). The crawler used up a lot of computer resources mostly CPU time, when it was run from one machine.

   (b) Distributed web crawler. The distributed crawler was developed to accommodate the immense demand for computer resources due to crawling huge websites. It was based on the standalone version of the web crawler.

2. Developing data analysing module

3. Developing data presentation module

Chapter 4

System analysis and design

4.1 System analysis

As indicated earlier, the system was developed in three major iterations. However, before the iterations had begun, initial requirement gathering and analysis was done. This covered the whole system. Initial requirement gathering and analysis Functional requirements: Nonfunctional requirements: Constraints ("Pseudo requirements"): Iteration 1 This iteration involved the development of the web crawler module of the system. Requirements Functional requirements crawl webpages in any domain given a base url take into account rich files i.e. pdf, doc,xls, discard similar pages resilient to network outages (saves state) Non-functional requirements minimal traffic not overload servers able to start/pause/resume/stop/resume on demand avoid getting banned Iteration 2 This iteration involved the development of the data analysis module. Iteration 3 This iteration involved the development of the data presentation module

Name	Create project
Id	1
Version	1
Summary	Create new webometric analysis project
Actors	User System
Entry conditions	User is registered User is logged in University domains are presented to user
Exit conditions	System starts data collection
Triggers	

Table 4.1: Use case 1

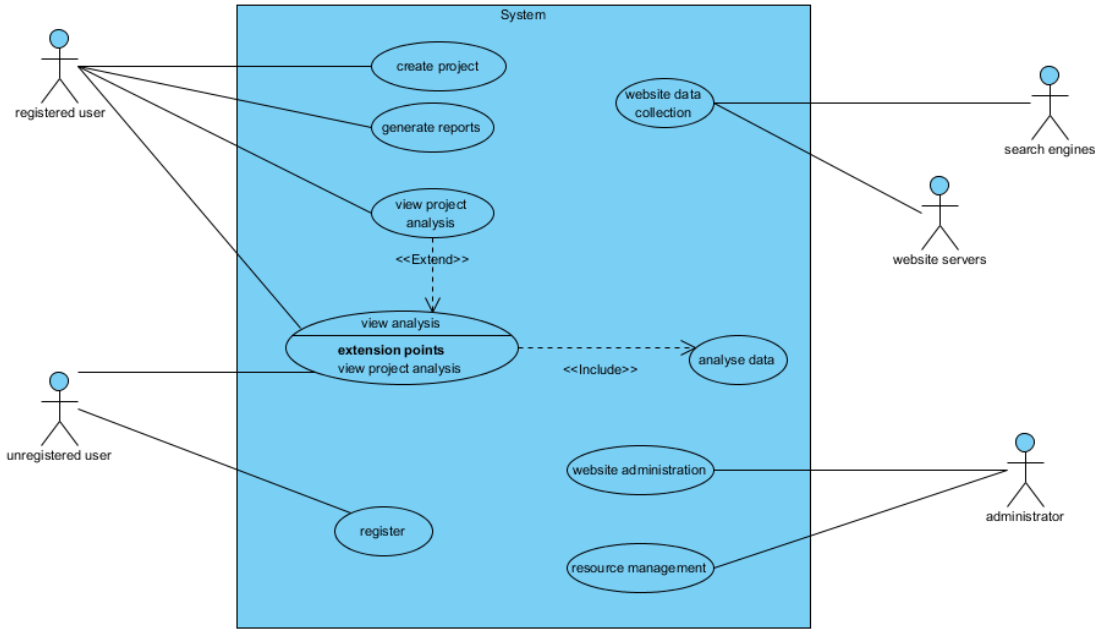

Figure 4.1: System use case model

4.1.1 Usecases

4.2 System design

Chapter 5

System implementation

Chapter 6

Results and Findings

Chapter 7

Conclusion

7.1 Recommendations

Chapter 8

References

Appendix A

User Manual

A.1 Using the system

A.2 Administation

Appendix B

Sample Code listing

```
"""
The MIT License

Copyright (c) 2014, mbacho (Chomba Ng'ang'a)

Permission is hereby granted, free of charge, to any person
    obtaining a copy
of this software and associated documentation files (the "
    Software"), to deal
in the Software without restriction, including without
    limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense,
    and/or sell
copies of the Software, and to permit persons to whom the
    Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
    included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
    EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
    EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
     OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
    ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
    DEALINGS IN
THE SOFTWARE.


file : walker.py
project : webometrics
```

```
"""

from re import sub

from random import randrange
from hashlib import sha256 as sh
from urlparse import (urlsplit, urlunsplit)
from scrapy.contrib.linkextractors.sgml import SgmlLinkExtractor
from scrapy.contrib.spiders import (CrawlSpider, Rule)
from ..items import WalkerItem
from .. import RICH_FILES


class Walker(CrawlSpider):
    name = 'walker'
    handle_httpstatus_list = [404, 500]
    jobid = None #use with scrapy server
    IGNORED_EXTS = [
        # images
        'mng', 'pct', 'bmp', 'gif', 'jpg', 'jpeg', 'png', 'pst',
            'psp', 'tif',
        'tiff', 'ai', 'drw', 'dxf', 'eps', 'svg',

        # audio
        'mp3', 'wma', 'ogg', 'wav', 'ra', 'aac', 'mid', 'au', '
           aiff',

        # video
        '3gp', 'asf', 'asx', 'avi', 'mov', 'mp4', 'mpg', 'qt', '
           rm', 'swf', 'wmv', 'm4a', 'mpeg',

        # other
        'css', 'exe', 'bat', 'bin', 'rss', 'zip', 'rar', 'xml',

        #script files
        'js', 'css', 'vbs', 'cs',
    ]

    DENY_DOMAINS = ['maktaba.ku.ac.ke', 'opac.mku.ac.ke', '
       library.kemu.ac.ke', 'opac.library.strathmore.edu']

    rules = (
        Rule(SgmlLinkExtractor(deny_extensions=IGNORED_EXTS,
           deny_domains=DENY_DOMAINS), callback='parse_item',
             follow=True,
             process_request='process_request', process_links='
                process_links', ),
    )
    start_urls = []
    allowed_domains = []

    def __init__(self, start, domain, jobid=None, *args, **
       kwargs):
```

```
        super(Walker, self).__init__(*args, **kwargs)
        if (type(start) is not str) and (type(start) is not
            unicode):
            raise TypeError('invalid type given for startpage')
        if (type(domain) is not str) and (type(domain) is not
            unicode):
            raise TypeError('invalid type given for domain')
        if start == '' or domain == '':
            raise ValueError('startpage or domain not provided')
        self.jobid = jobid
        self.start_urls = [start]
        self.allowed_domains = [domain]


    def parse_item(self, response):
        lnk = WalkerItem()
        lnk['status'] = response.status
        lnk['parent'] = response.request.headers.get('Referer',
            '')
        lnk['response_hash'] = '' if response.status != 200 else
            sh(response.body).hexdigest()

        type = response.headers['Content-Type']
        if ';' in type:
            type = type[:type.index(';')]
        lnk['type'] = type
        lnk['page'] = response.url
        return lnk

    def process_results(self, response, results):
        """
        This method is called for each result (item or request)
            returned by the spider, and it's intended to perform
        any last time processing required before returning the
            results to the framework core, for example setting
        the item IDs. It receives a list of results and the
            response which originated those results. It must
            return a
        list of results (Items or Requests).
        """
        return results

    def process_links(self, links):
        """called for each list of links extracted from each
            response using the specified link_extractor."""
        for link in links:
            split_link = urlsplit(link.url)
            link.url = urlunsplit((split_link.scheme, split_link
                .netloc, sub(r'//+', '/', split_link.path),
                                   split_link.query, split_link.
                                       fragment))
        return links

    def process_request(self, request):
```

```
        """
        called with every request extracted by this rule, and
           must return a request or None (to filter out the
           request)
        """
        ext = request.url.split(".")[-1]
        if ext in RICH_FILES.keys():
            request.method = 'HEAD'
        return request

    def is_valid_domain(self, domain):
        if domain != '':
            return True

        return False

    def is_valid_url(self, url):
        split_url = urlsplit(url)
        if split_url.scheme != '' and split_url.netloc:
            return True
        return False

    @property
    def user_agent(self):
        agents = [
            "chrome",
            "firefox",
            "opera",
            "safari",
        ]
        rand = randrange(0, len(agents))
        return agents[rand]
```

Listing B.1: Web crawler spider

Todo list