European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 03, 2020

Using Web Scraping In A Knowledge Environment To Build Ontologies Using Python And Scrapy

M. El Asikri[1], S. Krit[2], H.Chaib[3]

[1]*Department Mathematics and Informatics and Management, Laboratory of Engineering Sciences and Energy. Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University Agadir BP/638 Morocco*

Email: [1]meda.asikri@gmail.com, [2]salahddine.krit@gmail.com, [3]hchaib@gmail.com

Abstract : Web scraping, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

In this paper, among others kind of scraping, we focus on those techniques that extract the content of a Web page. In particular, we adopt scraping techniques in the Web e-commerce field. To this end, we propose a solution aimed at analyzing data extraction to exploiting Web scraping using python and scrapy framework .

Keywords: Web scraping , scrapy , e-commerce , data mining , data extraction , Crawlers , ontolgie .

1. INTRODUCTION

People might want to collect and analyse data from multiple websites. The different websites which belongs to the specific category displays information in different formats. Even with a single website you may not be able to see all the data at once. The data may be spanned across multiple pages under various sections. Most websites do not allow to save a copy of the data, displayed in their web sites to your local storage [1], the only option is to manually copy and paste the data shown by the website to a local file in the computer. This is a very tedious job which can take lot of time. Web Scraping is the technique which people can extract data from multiple websites to a single spreadsheet or database so that it becomes easy to analyse or even visualize the data. The aim of this study is to offers a review on web scraping techniques and software which can be used to extract data from web sites and also an application of framework scray to analyse knolegde from e-commerce web sites [2] .

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when you view the page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet, and so on. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and copy names and phone numbers, or companies and their URLs, to a list

(contact scraping). Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration.

Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. Because of this, tool kits that scrape web content were created. A web scraper is an Application Programming Interface (API) to extract data from a web site. Companies like Amazon AWS and Google provide web scraping tools, services and public data available free of cost to end users [3].

Newer forms of web scraping involve listening to data feeds from web servers. For example, JSON is commonly used as a transport storage mechanism between the client and the web server.There are methods that some websites use to prevent web scraping, such as detecting and disallowing bots from crawling (viewing) their pages. In response, there are web scraping systems that rely on using techniques in DOM parsing, computer vision and natural language processing to simulate human browsing to enable gathering web page content for offline parsing.

The aim of this paper is to give an overview of scraping techniques and also the application in e-commerce specially in the Moroccan case jumia.ma

2. STATE OF THE ART OF SCRAPING WEBSITES WITH FRAMWORKS

Existing approaches to implement a Web data scraper can be structured into two main categories: libraries for general-purpose programming languages, and frameworks.

2.1 Libraries

One of the most common approaches used by informaticians consists in constructing their own Web data scrapers using the programming language they are most familiar with. In this case, the logic behind the robot and the final result are implemented as conventional software programs, i.e. using the control and data structures of the language. Usually, third-party libraries grant access to the site by implementing the client side of the HTTP protocol, whereas the retrieved contents are parsed using built-in string functions, such as regular expression matching, tokenization and trimming. Third-party packages may also provide for more sophisticated parsing, such as HTML tree building and XPath matching.

One of the most popular site access libraries is libcurl [4] It supports the major features of the HTTP protocol, including SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form-based upload, proxies, cookies and HTTP authentication. Moreover, it has useful bindings to many programming languages.

Perl, which is one of the programming languages most widely used in bioinformatics, incorporates the Mechanize Web automation module .This module is able to interact with Web links and forms without additional parsing, and provides support for HTTPS, cookie management, HTTP authentication and history management, among others. Moreover, it allows the use of XPath through additional modules

In Java, the Apache HttpClient package emulates HTTP main features, i.e. all request methods, cookies, SSL and HTTP authentication, and can be combined with HTML parsing libraries such as jsoup [5] Java also supports XPath and provides several HTML cleaning libraries, such as htmlcleaner. Similarly, the BeautifulSoup is a Python HTML parsing library, which can be combined with language native support for HTTP connections. Moreover, in Unix-like environments, by simply piping operating system command-line

programs inside shell scripts, programmers are able to create Web data scrapers in one or few lines of code. Programs like curl (libcurl) and wget implement the HTTP client layer, while utilities such as grep, awk, sed and cut and paste can be used to parse and transform contents conveniently.In the case of server side robots, typically running inside Web applications, a 100% compatible technology with the programming language (typically PHP, Perl or Java) is recommended.

2.2.Frameworks

Using a general-purpose language to create robots has some drawbacks. Often, several libraries need to be integrated, in particular one for Web access and others to parse and extract content from the HTML documents. Furthermore, robots are known to be weak pieces of software, which are considerably affected by changes in the HTML of the accessed resources, and thus require continuous maintenance. In compiled languages, such as Java, any change in the implementation of the robot forces re-recompilation and even the re-deploy of the entire application.

Scraping frameworks present a more integrative solution. For example, Scrapy [6] is a powerful Web scraping framework for Python, where robots are defined as classes inheriting from BaseSpider class, which defines a set of 'starting urls' and a 'parse' function called at each Web iteration. Web pages are automatically parsed and Web contents are extracted using XPath expressions.

Other frameworks present domain-specific languages (DSL), which are specific programming languages designed for a particular domain and, therefore, robots are treated as independent and external artefacts. An example of this is Web-Harvest, a Web data scraping framework for Java. Here, Web extraction processes are described in XML (with the help of a visual environment) and are composed of several 'pipelines', which can include procedural instructions, such as variable definitions and loops, as well as many primitives, such as 'http' (to retrieve Web contents), 'html-to-xml' (to clean HTML) and 'xpath' to extract content. Another example of a Java Web data scraping framework is jARVEST, which also defines a DSL, but uses JRuby[7] for a more compact robot implementation.

Table 1 summarizes and compares several of the most popular open-source Web scraping libraries and frameworks.

	Type C: HTTP client P: Parsing F: Framework	Domain-specific language	**API/stand alone**	Language	Extraction facilities R: Regular expressions H: HTML parsed tree X: XPath C: CSS selectors
UNIX shell	CP	No	SA	bash	R
Curl/libcurl	C	No	Both	C + bindings	
Web-Harvest	F	Yes	Both	Java	RX
Jsoup	CP	No	API	Java	HC
HttpClient	C	No	API	Java	
jARVEST	F	Yes	Both	JRuby/Java	RXC

WWW::Mechanize	CP	No	API	Perl	RX
Scrapy	F	No	Both	Python	RX
BeautifulSoup	P	No	No	Python	H

Table 1: Comparison of open source web scraping libraries and frameworks

We have selected several available Web scraping packages oriented to programmers. There are six libraries implementing an HTTP client (C) and/or HTML parsing/extraction (P) and three frameworks (F). Web-Harvest and jARVEST frameworks present a domain-specific language for defining robots, based on XML and Ruby, respectively. For all the analyzed alternatives, we report their extraction facilities, including regular expressions (R), HTML parsed tree (H), XPath expressions (X) and CSS Selectors (C).
As we see in section 2various Open Source Crawlers differ from one another in terms of scalability, flexibility and their performance in different scenario. Adaptation of particular crawlers by user or organization totally depends on their requirement. There are some key feature differentiation is given in Table 1 which will help the user to select the appropriate crawler according to their requirement .

Basic Definitions
Technique of web scaping :
Web scraping is the process of automatically mining data or collecting information from the World Wide Web. It is a field with active developments sharing a common goal with the semantic web vision [8], an ambitious initiative that still requires breakthroughs in text processing, semantic understanding, artificial intelligence and human-computer interactions. Current web scraping solutions range from the ad-hoc, requiring human effort, to fully automated systems that are able to convert entire web sites into structured information, with limitations.

a.HTML parsing :
Many websites have large collections of pages generated dynamically from an underlying structured source like a database. Data of the same category are typically encoded into similar pages by a common script or template. In data mining, a program that detects such templates in a particular information source, extracts its content and translates it into a relational form, is called a wrapper[9]. Wrapper generation algorithms assume that input pages of a wrapper induction system conform to a common template and that they can be easily identified in terms of a URL common scheme.[10] Moreover, some semi-structured data query languages, such as XQuery and the HTQL, can be used to parse HTML pages and to retrieve and transform page content.

b. DOM parsing:
By embedding a full-fledged web browser, such as the Internet Explorer or the Mozilla browser control, programs can retrieve the dynamic content generated by client-side scripts. These browser controls also parse web pages into a DOM tree, based on which programs can retrieve parts of the pages.

c. Vertical aggregation:
There are several companies that have developed vertical specific harvesting platforms. These platforms create and monitor a multitude of “bots” for specific verticals with no "man in the loop" (no direct human involvement), and no work related to a specific target site. The preparation involves establishing the knowledge base for the entire vertical and then the

platform creates the bots automatically. The platform's robustness is measured by the quality of the information it retrieves (usually number of fields) and its scalability (how quick it can scale up to hundreds or thousands of sites). This scalability is mostly used to target the Long Tail of sites that common aggregators find complicated or too labor-intensive to harvest content from.

d. Semantic annotation recognizing:
The pages being scraped may embrace metadata or semantic markups and annotations, which can be used to locate specific data snippets. If the annotations are embedded in the pages, as Microformat does, this technique can be viewed as a special case of DOM parsing. In another case, the annotations, organized into a semantic layer,[11] are stored and managed separately from the web pages, so the scrapers can retrieve data schema and instructions from this layer before scraping the pages.

e. Computer vision web-page analysis:
There are efforts using machine learning and computer vision that attempt to identify and extract information from web pages by interpreting pages visually as a human being might.[12]

Web scraping
Nowadays, quite a lot of researchers are working on extracting information about types of events, entities or relationships from textual data. Information extraction is used for search engines, news libraries, manuals, domain-specific text or dictionaries. A form of information extraction is text mining, an information retrieval task aimed at discovering new, previously unknown information, by automatically extracting it from different text resources . In information extraction, text mining is used to scrap relevant information out of text files by relying on linguistic and statistic algorithms (Fig 1).

Fig 1 : Structure of the Web Scraping

Web search and information extraction is typically performed by Web crawlers. A Web crawler is a program or automated script that browses the WWW in a methodical, automated manner[13] . A more recent variant of Web crawlers are Web scrapers, which are aimed at looking for certain kinds of information–such as prices of particular goods from various online stores—extracting, and aggregating it into new Web pages .
Scrapers are basically adopted to transform unstructured data and save them in structured databases. In screen scraping, a special form of scraping, a program extracts information from the display output of another program . So that, the output which is scraped is created for the end user and not for other programs that is the difference to a normal scraper. In this paper, we focus on Web scrapers that extract textual information from Web pages. There are many methods to scrap information from the Web. Since barriers to prevent machine automation are not effective against humans, the most effective method is human copy-paste. Although sometimes this is the only way to export information from a Web page, this is not feasible in practice, especially for big company projects, being too expensive. Another method is text grepping in which regular expressions are used to find information that

matches some patterns. Further Web scraping techniques are HTTP programming, DOM parsing, and HTML parsers. Finally, a Web scraping method consists of making scraper sites that are automatically generated from other Web pages by scraping their content .It is worth noting that Web scraping may be against the terms of use of some websites. Being interested in the scientific issues concerned with the adoption of Web scraping to perform Web advertising, in this paper we do not take into account legal issues on adopting and implementing Web scraping techniques.

Scraping e-commerce web site with scrapy and python :

A big percent of the world's data is unstructured, estimated around 70%-80%. Websites are a rich source for unstructured text that can be mined and turned into useful insights. The process of extracting information from websites is usually referred to as Web scraping [14].

There are several good open source Web scraping frameworks, including Scrapy , Nutch [15] and Heritrix [16] .

For medium sized scraping projects, Scrapy stands out from the rest since it is:

- Easy to setup and use
- Great documentation
- Mature and focused solution
- Built-in support for proxies, redirection, authentication, cookies and others
- Built-in support for exporting to CSV, JSON and XML

In this section, we will try to extract information from web site e-commerce (www.jumia.ma) and analyzing it in order to build and ontologies of products classification:

a. Using Scrapy

There are various methods to use Scrapy , it all depends on your use case and needs, for example:

- Basic usage: create a Python file containing a spider. A spider in Scrapy is a class that contains the extraction logic for a website. Then run the spider from the command line.
- Medium usage: create a Scrapy project that contains multiple spiders, configuration and pipelines.
- Scheduled scraping: use Scrapyd to run scrapy as a service, deploy projects and schedule the spiders.
- Testing and debugging: use Scrapy interactive shell console for trying out things

In this section , we will focus on running spiders from the command line, since all other methods are similar and somehow straightforward.

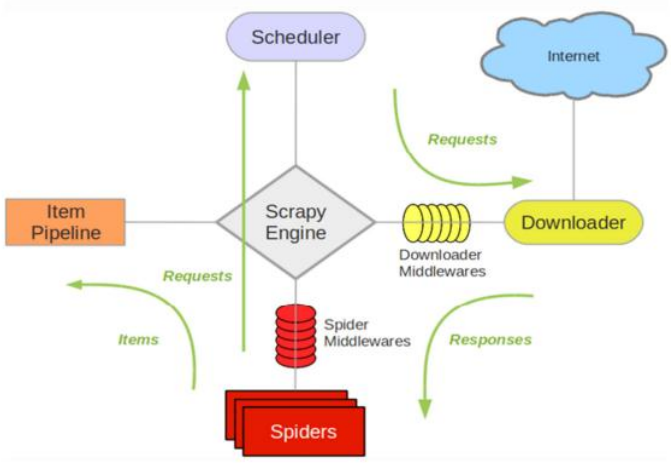

Fig 3 : System architecture of scrapy

b. Understanding Selectors

Before scraping the website, it is important to understand the concept of selectors in scrapy. Basicly, selectors are the path (or formula) of the items we need to extract data from inside a HTML page

We can either use the CSS or XPath selectors to point to the items you want to extract. we prefer the CSS selectors since they are easier to read. To find the CSS/XPath of items we greatly recommend to install the Chrome plugin SelectorGadget. On Firefox we can install Firebug and Firefinder.

e. Example of scraping e-commerce web site

This is the code used to scrape the web site :

```
import scrapy

class Jumia.maSpider(scrapy.Spider):
   name = "Jumia.ma"  # Name of the Spider, required value
   start_urls = ["http://www.jumia.ma. /"]  # The starting url, Scrapy will request this URL in parse

   # Entry point for the spider
   def parse(self, response):
      for href in response.css('.sp-cms_unit--ttlattr(href)'):
         url = href.extract()
         yield scrapy.Request(url, callback=self.parse_item)

   # Method for parsing a product page
   def parse_item(self, response):
      original_price = -1
      savings=0
      discounted = False
      seller_rating = response.css('.vip-product-infoats .inline-block small::text'). extract()[0]
      seller_rating = int(filter(unicode.isdigit,seller_rating))

      # Not all deals are discounted
      if response.css('.vip-product-infobhead::text').extract():
         original_price = response.css('.vip-product-infobhead::text').extract()[0].replace("DH", "")
         discounted = True
         savings = response.css('.vip-product-infossage .noWrap::text').extract()[0].replace("DH", "")
      yield {
         'Title': response.css('.product-title:text').extract()[0],
         'Category': response.css('.product-title span a+ a::text').extract()[0],
         'OriginalPrice': original_price,
         'CurrentPrice': response.css('.vip-product-infoice::text').extract()[0].replace(u"\xa0", ""),
         'Discounted': discounted,
         'Savings': savings,
         'SoldBy': response.css('.vip-product-infoats a::text').extract()[0],
         'SellerRating': seller_rating,
         'Url': response.url
```

```
}
```

The previous code will request the deals page at Jumia.ma, loop on each product link and extract the required data inside the parse_item method.
To run the spider we are going to use the runspider command, issue the following command:

scrapy runspider Jumia.ma_spider.py -o deals.csv

When the extractions finishes we will have a CSV file (deals.csv) containing Jumia.ma deals for the day, contains a sample out. You can also change the format of the output to JSON or XML by changing the output file extension in the runspider command, for example, to get the results in JSON issue:

f. Data Analysis
Looking at the data of 300 product deals, I came with the following analysis:

i. Top Categories
There are 113 different categories in the deals data. I found it a bit surprising that the top two deals categories are Watches (10%) and Handbags (9%) as shown in Fig [3]. But if we combine Mobile Phone (7.6%), Laptops (5.6%) and Tablets (4%) into a general Electronics category, we get (17%), which bring it to the top category and makes much more sense.

Category	Count	Percent
Watches	120	10.017%
Handbags	108	9.015%
Mobile Phones	92	7.679%
Laptops	67	5.593%
Perfumes	63	5.259%
Eyewear	59	4.925%
Tablets	48	4.007%
Tops	38	3.172%
Toys	37	3.088%
Makeup	35	2.922%

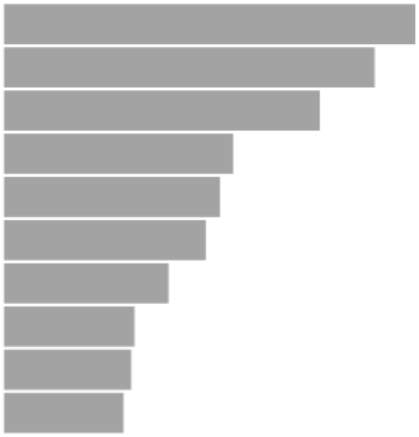

Fig 3: Top Product Categories in Deals

ii. Top Sellers
This data is more interesting, from the 86 different sellers, around 78% of the 1,200 deals are provided by one seller . Also, as illustrated in Fig [4], the next seller accounts for only 2%. The best seller deals are in 98 different categories (87% of categories) ranging from Baby Food, Jewelry to Laptops and Fitness Technology. In comparison, JUMBOELECTRONICS deals are in 3 related categories.One can conclude that the first seller is somehow related to Jumia.ma or a very active seller!

Seller	Count	Percent
dod	928	77.462%
JUMBOELECTRONICS	25	2.087%
faisaldurvesh	15	1.252%
IT-store	13	1.085%
user-AMCMUK	11	0.918%
FINEDEALS	10	0.835%
Luxury-Beauty	10	0.835%
DJperfumes	10	0.835%
For-value-for-pleasure	8	0.668%
wmm891009	7	0.584%

Fig 4: Top sellers in deals

Iii. Sellers Ratings

There are 14 sellers with 0% Positive Rating, we don't know what does that mean in Jumia.ma terminology. It might be that these sellers don't have a rating yet or their overall rating is too low, so it was decided to hide it. The average rating for the remaining non-zero rating is 86.542 with a standard deviation of 8.952. There does not seem to be a correlation between seller rating and discount percentage as shown in Fig [5], it seems other variables such as service and quality factors more in rating than discount percentage.

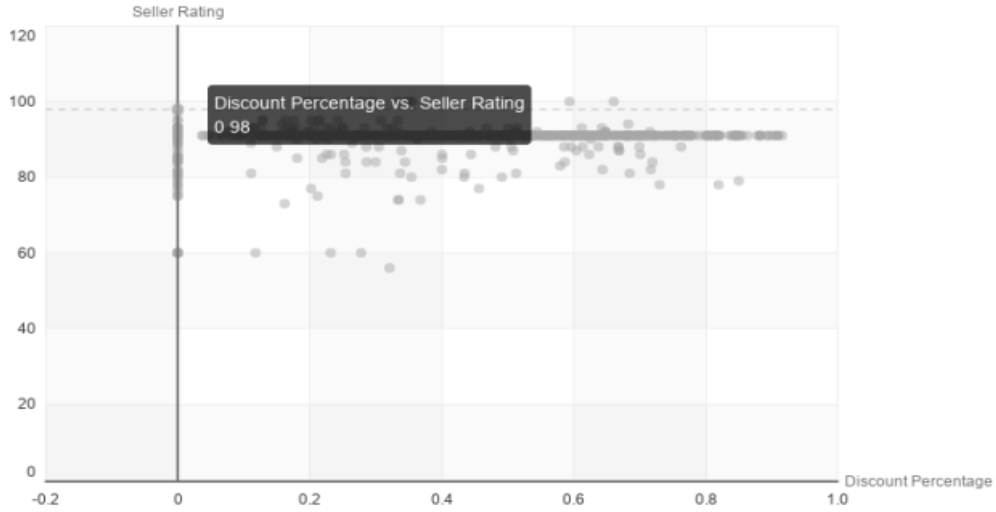

Fig 5: Discount vs rating

iv.Product Prices and discounts

We can get a lot of useful information looking at the deals prices and discounts, for example:The average discount percentage for all the deals is 42% .If we have the urge to buy something from Jumia.ma after scraping and analyzing the deals, but you are on a tight budget, you can always find the cheapest item .

3. CONCLUSION & DISCUSSION

In this paper, we presented a comparative studies of scraping methods according to it, we identified most of the web scrapers are often quite generic and mostly designed to perform common, simple tasks. In other words, they may appear not to be as flexible and universal as you would expect. All the web scraper developers try to make their products scrape all kinds of web pages, but we realized some web scraping software are better suited for one type of

task and some are suited for another, and also we presented a solution based on scrapy framework for scraping e-commerce web site in order to make betters decision in deals case and to extracting structured data in order to make better decision.

4. REFERENCES

[1] Penman, R.B., Baldwin, T., Martinez, D., 2009. Web scraping made simple with site scraper. Text.
[2] M. El Asikri , S. Krit , H.Chaib a brief survey of creating semantic web content with protégé international journal of engineering, science and - Volume 7, Issue 3, March 18 Pages: 131-140
[3] Le QT, Pishva D (2015) Application of web scraping and Google API service to optimize convenience stores' distribution. In: 17th International conference on advanced communication technology (ICACT). pp 478–482
[4] Pei Xia, Makoto Matsushita, Norihiro Yoshida, Katsuro Inoue Studying reuse of out-dated third-party code in open source projects Information and media technologies 9(2) : 155-161 2014 Japan Society for Software Science and Technology
[5] P Houston - Instant jsoup How-to 2013 - books.google.com
[6] http://scrapy.org
[7] Charles O. Nutter ,Thomas Enebo ,Nick Sieger, Ola Bini ,Ian Dees "Using JRuby: Bringing Ruby to Java 1st Pragmatic Bookshelf ©2011 ISBN: 9781934356654
[8] M. El Asikri, S. Krit and H.Chaib Development of Semantic Web applications: state of art and critical review ICEMIS '18 Proceedings of the Fourth International Conference on Engineering & MIS 2018 Article No. 55 Istanbul, Turkey — June 19 - 20, 2018 ACM New York, NY, USA ©2018 table of contents ISBN: 978-1-4503-6392-1 doi>10.1145/3234698.3234753
[9] M. El Asikri , S. Krit , H.Chaib , M. Kabrane, H. Ouadani, K. Karimi,K.Bendaouad and H. Elbousty Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University, Agadir : Mining the Web for learning ontologies : state of art and critical review, (ISCSA2017 submission 45) à Errachidiya .
[10] Song, Ruihua; Microsoft Research (Sep 14, 2007). "Joint Optimization of Wrapper Generation and Template Detection" (PDF). The 13th International Conference on Knowledge Discovery and Data Mining.
[11] http://www.gooseeker.com/en/node/knowledgebase/freeformat
[12] Roush, Wade (2012-07-25). "Diffbot Is Using Computer Vision to Reinvent the Semantic Web". www.xconomy.com. Retrieved 2013-03-15.
[13] Trupti V. Udapure1 , Ravindra D. Kale2 , Rajesh C. Dharmik3 Study of Web Crawler and its Different Types IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05 www.iosrjournals.org
[14] M. El Asikri , S. Krit , H.Chaib A BRIEF SURVEY OF CREATING SEMANTIC WEB CONTENT WITH PROTÉGÉ INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND - Volume 7, Issue 3, March 18 Pages: 131-140
[15] José E. Moreira , Maged M. Michael , Dilma Da Silva , Doron Shiloach , Parijat Dube, Li Zhang , Scalability of the Nutch search engine ICS '07 Proceedings of the 21st annual international conference on Supercomputing Pages 3-12 Seattle, Washington — June 17 - 21, 2007 ACM New York, NY, USA ©2007 ISBN: 978-1-59593-768-1 doi>10.1145/1274971.1274975
[16] G. Mohr, M. Stack, I. Rnitovic, D. Avery, and M. Kimpton. Introduction to Heritrix. In 4th International Web Archiving Workshop, 2004