IEEE-SEM, Volume 11, Issue 8, August-2023
ISSN 2320-9151

Automatic Scientific Literature Gathering and Analysis from Textual Corpus using Web Scraping and Locality Sensitive Hashing

Asfandyar Ahmed
Department of Computing and Technology
Abasyn University
Peshawar, Pakistan
supremeconsole33@gmail.com

Dr.Abdus Salam
Department of Computing and Technology
Abasyn University
Peshawar, Pakistan
dr.salam@abasyn.edu.pk

***Abstract*—Scientific internet repositories are central cyber places where papers that are scholastic saved and maintained. Aided by the nature associated with the unstructured and information/metadata that is semi-structured these repositories, literary works analysis for scholar writing becomes a challenge State-of-the-art web scraping tools are capable of parsing HTML, XML and JSON files. Wrapper induction consists of three contributions. In web data processing wrapper is a program that extracts content from a source and translates it into a way of words. By using the supervised learning, the wrapper induction learns the info extraction rules from manually labeled training examples. Generally, the web data is scrapped utilizing HTTP links or through an updated web browser. Text similarity decides how adjacent two chunks of text are both in surface nearness and meaning. HLRT is intended a twin of finite state automata induction so as to create wrapper induction faster. Semantic and syntactic web scraping are the two categories of web scrapping techniques. The aim of this paper is to allow scrapers to automatically collect and analyze scientific literature using web scraping from research repositories and using the acquired data to discover most relevant research papers**

Keywords— web scraping, web crawling, text mining, relevant literature search, bypass 504 error, splash, web scraping techniques, locality sensitive hashing,

I. INTRODUCTION

In this world, the technological advancements are changing our lives rapidly. This development has characterized an exponential growth in the research output. Every year over 1.8 million scientific research papers are being published in about 28000 research journals [1]. Researchers often face many problems on their path of publishing and presenting their works to universities or scientific journals. Acquisition of related research articles by a research scholar is an exhaustive ordeal to keep up to pace with the ever-increasing body of knowledge. Academicians must broadly search repeatedly the textual corpus for scientific literature gathering and it is very time consuming process when performed manually. When researchers visit different research repositories to find related work using key words, the process produces a large amount of scientific literature. Of course selecting a few related articles out of thousands of research papers found by the search engines is a tedious job to go through. Considering this massive scale of text data there is a dire need for a user-friendly solution that may allow user to select and download the most relevant articles in a short time and organize them in a manner that may allow researchers to perform literature survey in least amount of time.

Web scraping is widely acknowledged as an efficient and powerful approach for collecting large amount of data (Mooney et al. 2015) [2]. Scraper extracts data from the Web and stores into the item pipelines for further inspection. Generally, the web data is scrapped utilizing HTTP links or through an updated web browser. The process of scraping information from the Internet can be divided into two steps; obtaining web resources and then extracting desired data from the acquired data. This can be accomplished either manually or automatically by a web crawler.

Present day web scraping methods have become over complicated from smaller ad hoc, human-aided procedures to fully automated systems that are able to convert entire websites into well-organized data set. State-of-the-art web scraping tools are capable of parsing HTML, XML and JSON files (Butler 2007) [3] . Beautiful Soup is actually just a simple content parser. It can't do much else, as it requests library to actually retrieve the web page for it to scrape. Whereas, Scrapy is an entire framework consisting of many libraries, capable of retrieving, parsing and extracting data from a web page all by itself.

Then comes the need to analyze and summarize the data collected through web scrapping techniques. Text similarity decides how adjacent two chunks of text are both in surface nearness and meaning [4]. The degree of similarity between the resources are indicated by comparing the text with available resources through text similarity measurement. Many studies are carried out for finding text similarity and resulting in different algorithms and techniques.

II. LITERATURE SURVEY

The World Wide Web is composed of HTML and CSS documents characterized as a visual Web. Web browsers use HTML and CSS data to display these documents. Web scraping is a process of gathering unstructured data from the (WWW) World Wide Web. Semantic and syntactic web scraping are the two categories of web scrapping techniques. Syntax related web scraping collects information from the website by parsing website components and web programing language data. The expository properties of parsing Hypertext Markup Language (HTML), essentials that are connected to the components in term of Cascading Style Sheets (CSS) selectors is a technology that provides the extraction of data after selection.

In web data processing wrapper is a program that extracts content from a source and translates it into a way of words. Wrapper induction extracts the information from specific

HTML pages and automate data extraction, which focuses on repetitive data, and collection tasks are the two main techniques of wrapper generation [5]. By using the supervised learning, the wrapper induction learns the info extraction rules from manually labeled training examples [4]. Wrappers were constructed for the aim of content extraction. it's constructed by many researchers but only experts can construct wrappers because it's awfully complex task [6] [7] [8]. The wrapper must be rebuild after the page is modified. Additionally, wrapper maintenance may be a difficult task.

Kushmerick et al. [6] introduced wrapper induction, which could be a coded method for content extraction. Wrapper induction consists of three contributions. First, the wrapper construction problem is formalized as induction. Second, it defines HLRT class, which is chargeable for handling various Internet information resources, where their contents are displayed in tabular layouts. HLRT is intended a twin of finite state automata induction so as to create wrapper induction faster. HLRT class is efficiently learnable and since of that it'll enhance the extraction. Third, heuristic knowledge is employed for composing algorithm's oracle where oracle is that the key on induction. This method is extremely time consuming which affects its efficiency [9].

Text similarity methods play important role in tasks like data retrieval, document classification, clustering, and detection of subject, generation of questions, answering the questions, short answer scoring, text summarization. Discovering similarity between documents is a crucial part of text similarity that's used as a primary function for locating sentence, paragraph and document similarities [10]. Text are often similar in two ways lexically and semantically. Lexically two words are similar if they have same character sequence. Two words are semantically equal if they represent the identical thing, utilized within the identical way, utilized within the identical context and one can be a sort of another. Knowledge-Based similarity may well be a semantic similarity measure that determines the degree of similarity between words using information derived from semantic networks [10].

Today's world is driven by the information and the Internet is a main source of information. Researchers rely on the World Wide Web to collect research papers that are vital for their research work. Researchers regularly search study and collect related research papers from various web repositories and when performed manually, this is often a time consuming, error- prone process. Therefore, it is very important to create a simplified framework that can easily find related papers from web and organize the extracted data in desired manner.

To know how the data extraction process has evolved has so much one must understand the techniques involved in this method of web scraping is important scraping has been around nearly as long as the web [11]

III. OBJECTIVES

The objective is to allow users for simple and quick literature analyses. The overall objectives of this research work are stated below:

- To automatically collect and analyze scientific literature from web repositories such as Google Scholar using web scraping techniques and select most relevant research papers
- Transform unstructured data acquired from the web into structured format and support the scrapping of https (secured site) sites plus Ajax enabled sites too.
- To identify best web scraping practices to follow without being blocked by various security mechanisms.
- To study the application of locality sensitive hashing text similarity algorithm for selection of desired number of most relevant research papers.

IV. IMPLEMENTATION AND METHODOLOGY

XP Path

Along with other selectors like CSS XPATH has become the most powerful query language used by web scrapers to select nodes from an XML or HTML documents. It stands for XML path language and is used in many aspects of web development. An example on an operating system when a scraper want to navigate from point A to Point B. It will pass by some subfolders between two sub folders. There is a slash separating them.

This spot syntax used by an operating system Mac OS or Linux is pretty similar to the one used by XPATH now XPATH has three versions XPATH 1.0 2.0 and XPATH 3.1 which is the latest one. Unfortunately Scrapy or web browsers support only version 1. So it should be made sure when reading articles or documentations that it should be read about the supported version Now to coders HTML document is just lines and lines of code for XPATH or any other selectors the HTML document is seen like a tree that has different kinds of nodes XPATH and HTML document or XML document can distinguish between 7 kinds of nodes.

But in the reality a scraper will come across four of them element nodes, attribute nodes, comment nodes and text nodes an element node represents the HTML tag like the paragraph tag the Header one tag and the div tag the attribute node represents an attribute within an element node like the href attribute the id attribute and the class attribute. Next is the command node. It's the comment that is written on an HTML web page. And finally the text nodes that represents the text content or the value within an element node.

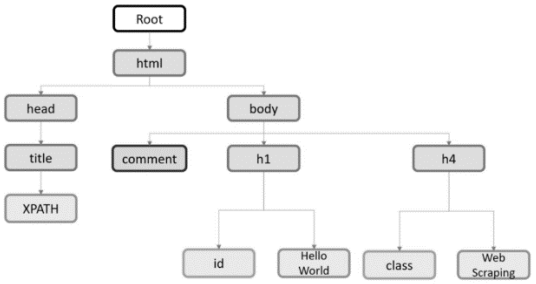


Fig 1. XP Path Syntax

Now an Html document is seen by XPATH as a tree similar to a family tree but this one is often called the document object model that contains all kinds of nodes that are part of the HTML page. The only node that is not part of the HTML document is the root node because it is dealt with a tree it would build relationships between nodes and they want to understand and memorize all of them because they are the key to write valid XPATH expressions.

Relationships of Nodes -> Parent

So the first relationship is called Parent the body node is the parent of the comment node the h1 node and the h4 node.

Relationships of Nodes -> Children

Then we have the children relationship. For example the comment node the h1 node and the h4 node are all children of the body note.

Relationships of Nodes -> Siblings

Then we have siblings and siblings means all the nodes that has the same parent example the head and the body are siblings because they share the same parent which is the html node.

Relationships of Nodes -> Ancestors

Next we have ancestors which means no parent grandparents and grand grandparent for example the ancestors of the h1 node are the body node the html node and the root node.

Relationships of Nodes -> Descendants

Now the last one is called descendant which means no children and children's children. It's like the inverse of the ancestor's relationship example the descendants of the body node are all its children. They comment node the h1 node the h4 node including the H-1 note and the h4 node children.

Scrapy Shell

The Scrapy shell is an interactive shell where the scraping code is debugged very quickly, without having to run the spider. It's meant to be used for testing scraper code, but it can be used to test any kind of code as it is also a regular Python shell.

The shell is used for testing XPath or CSS expressions to see how they work and what data is extracted from the website that is needed to be scraped. It allows the scraper to interactively test the expressions of a scraper while writing the spider, without having to run the spider to test every change.

For developing and debugging the spiders the Scrapy shell is an invaluable tool If IPython installed on the operating system, the Scrapy shell will use it (instead of the standard Python console). Smart auto completion and colorized output among other things is by the powerful IPython console.

It is highly recommended to install IPython, especially for Unix systems (where IPython excels).

Scrapy

Scrapy, overall, is a web crawling framework written in Python. One of its main advantages is that it's built on top of Twisted, an asynchronous networking framework, which in other words means that it's: a) really efficient, and b) Scrapy is an asynchronous framework.So, to illustrate Scrapy is supported under or uses Python 2.7 and Python 3.3. So a scraper can pretty much, depending on the version of Python, then the scraper is pretty much good to go. So Python 2.6, important thing to note support was dropped starting at Scrapy 0.20. So scrapers have to bear that in mind, and Python 3 support was added in Scrapy 1.1 with each version of Python the support will be added in Scrapy.

The Engine gets the demands which are initial crawl from the Spider. The Engine schedules the demands in the Scheduler and wants the requests which are next crawl. The Scheduler comes back the requests which can be next the Engine. The demands are sent by the Engine to the Downloader, passing through the Downloader Middlewares (see process_request()). Once the page completes getting the Downloader yields a reply (with that web page) and delivers it to the Engine, moving through the Downloader Middlewares (see process_response()). The Engine gets the Response from the Downloader and sends it towards the Spider for processing, passing through the Spider Middleware (see process_spider_input()). The Spider processes the Response and returns scraped items and needs that are newto follow) to your Engine, passing through the Spider Middleware (see process_spider_output()). The Engine delivers prepared items to Item Pipelines, then deliver processed needs to the Scheduler and asks for possible needs that are next crawl. The process repeats (from step one) until there are no more demands through the Scheduler.

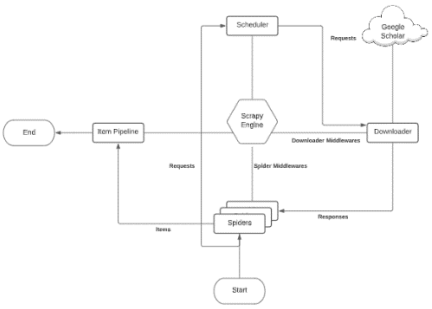


Fig 2. Scrapy Architecture

Splash

Splash is like a lightweight browser. The way we interact with that browser is by writing some code that splash can understand not by using icons like Chrome for example and it's meant to be used with scrapy. JavaScript requires an engine to be executed.So I'm gonna tell you what engine each browser use including splash so Chrome has what we call the V8 engine. Firefox has spider monkey Safari has Apple Web Kit.That's the same engine used by splash and Microsoft Edge has Chakra.

The state-of-the-art that is current scraping solutions and technologies give a wide variety of simple user friendly tools. These solutions came a very way that is long a past that has been crowded by complexities impeding content extraction; especially for normal skilled users. A majority of these methods relied on API and/or techniques that are code-based extract content from designated web sites. The complexity that is traditional in working with these APIs and code-based structures could be explained into the convoluted technicalities behind the three actions in the process. For the provided user to clean or crawl content he/she would will often have to code that is difficult parameters starting with the site URL become crawled ("Site Access"). The process of automatic content extraction (well regarded as "Web Scraping", a.k.a "Web Crawling") happens over three broad actions: (1). Site Access, (2). HTML parsing and contents removal and (3). Output building [12].The situation here is based on the possible lack of dynamic projects of URL parameters provided the understood undeniable fact that site URLs are often difficult coded. This very problem blows out of proportion, particularly in the following stage of "HTML parsing and extraction" that is content. HTML parsing and content removal usually takes spot by analyzing the physiology of web-sites therefore the content that is underlying. An instance that is conventional in which a

individual would need to realize the xPath structure for the web-site along with know its name in order to include/exclude it for content extraction [13].

Nowadays most of the browsers provide integrated development tools that are designed to assist web designers to flick through the dwelling of offered web page. The value of this development tools is unquestionably quite high for web-developers. Nevertheless, in regards back into the way it is of extracting web content from medical repositories the process continues to be complex because of the xPath that is hierarchical depicted in Figure 3a,b.

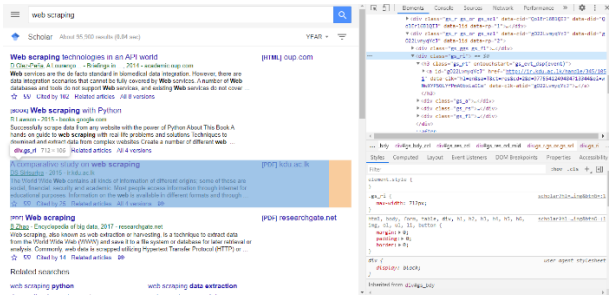
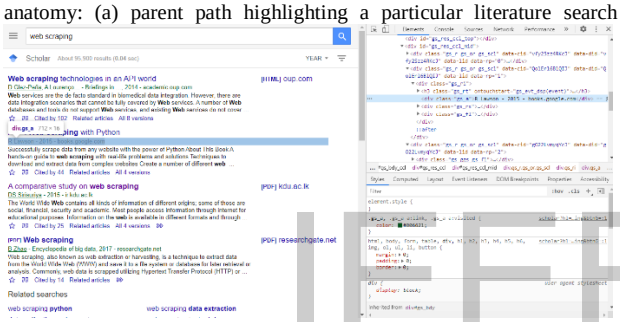
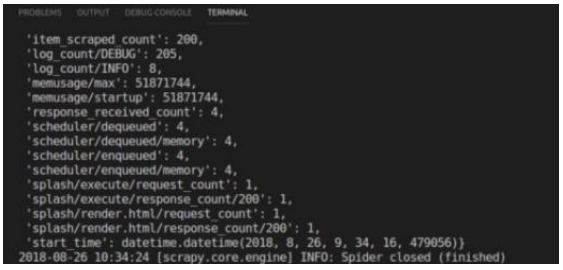


Figure 3. Google Chrome's Inspect Element panel dissecting a webpage anatomy: (a) parent path highlighting a particular literature search hit



(b) Highlighting a sub-component of the parent search hit.

Over time, many of the web scraping tools identified in literature [14] had been simplified to be able never to just offer free resources, but also to code solutions which can be free. Scrapy, Splash and Lua was indeed utilized to conquer the complexities to automatically collect and analyze scientific literature from web repositories such as Google Scholar.



```
'item_scraped_count': 200,
'log_count/DEBUG': 205,
'log_count/INFO': 8,
'memusage/max': 51871744,
'memusage/startup': 51871744,
'response_received_count': 4,
'scheduler/dequeued': 4,
'scheduler/dequeued/memory': 4,
'scheduler/enqueued': 4,
'scheduler/enqueued/memory': 4,
'splash/execute/request_count': 1,
'splash/execute/response_count/200': 1,
'splash/render.html/request_count': 1,
'splash/render.html/response_count/200': 1,
'start_time': datetime.datetime(2018, 8, 26, 9, 34, 16, 479056)}
2018-08-26 10:34:24 [scrapy.core.engine] INFO: Spider closed (finished)
```

Fig 4(a) . Automated pagination

Automated Pagination, also referred to as automatic paging, is being carried well this is the procedure of dividing a document into discrete pages, this means bundle of data on different web page. The pages had various unique urls. So scraper took these urls one by one and cleaned these pages.

But a scraper should keep in mind is when scrapers stops pagination. Generally speaking pages have actually next switch, this switch that is next able plus it get disable whenever pages are finished. This technique is employed getting url of pages till the page that is next is ready when it get disable no page is kept for scraping.

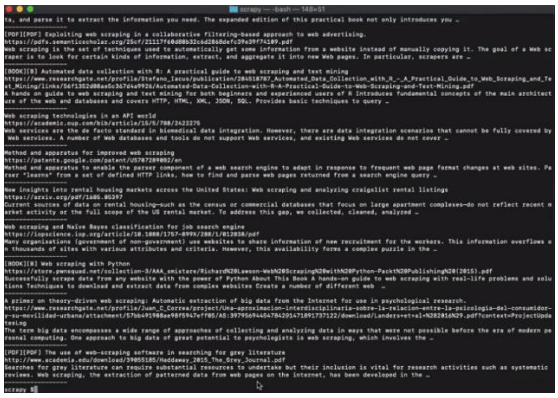


Fig 4(b). Result of scrpaed articles from google scholar

The unstructured data will be saved in to CSV file by running the spider with the crawl command and export the results to a csv file. Now when a scraper hits the goggle scholar site or any another site for scraping the data the site will ban that scraper.

Best web scraping practices to follow without being blocked by various security mechanisms.

First, a scraper have to understand what is robots.txt file and what's its functionality. So, basically it tells internet search engine crawlers which pages or files the crawler can or can't request from your website. This is certainly mainly utilized in order to avoid overloading any internet site with requests. This file provides guidelines which can be standard scraping. Numerous websites allow GOOGLE to allow them clean their websites. One can find robots.txt file on websites — http://example.com/robots.txt.

Often websites that are specific User-agent: * or Disallow:/ within their robots.txt file this means they don't want you to definitely scrape their sites.

Essentially device that is anti-scraping for a fundamental guideline which will be: Is it a bot or a individual?

```
User-agent: *
Allow: /
Disallow: /connector/
Disallow: /plugins.
Disallow: /firststeps.
Disallow: /publicliterature.PublicLiterature.search.html
Disallow: /lite.publication.PublicationRequestFulltextPromo.requestFulltext.html
Disallow: /amp/authorize
Allow: /signup.SignUp.html
Disallow: /signup.
```

Fig 5. Google Scholar robot.txt file

Here is the means that is easiest for anti-scraping mechanisms to caught a scraper red-handed. Scraper will be obstructed in the event that it will keep utilizing the exact same internet protocol address for each and every request. So, for every scraping that is prosperous, scraper have to use a new IP for every demand. A pool must be had by scraper of at the least 10 IPs prior to making an HTTP request. To avoid getting blocked scraper should use rotating that is proxy like Scrapingdog or every other Proxy services

Another technique to avoid getting banned by websites is by using fake user agents the User-Agent demand header is a character string that allows servers and community peers

identify the application form, operating system, vendor, and/or form of the user representative that is asking for. Some web sites block particular requests if they contain User-Agent that don't fit in with a web browser that is major. If user-agents aren't set sites which are many enable viewing their content. Scraper will get it's user-agent by typing what's my user representative on goggle. Somewhat technique that is same employed by an anti-scraping system which they use while banning IPs. Scraper will be banned right away if scraper work with exactly the same user-agent for every request. What's the solution? Well, the perfect solution is is pretty simple scraper need to either develop a range of user-Agents or usage libraries like maybe fake-user agents.

Another technique to avoid getting banned by websites since the rate is well known scraper of crawling the internet sites by humans and bots is extremely various. Bots can clean sites at a really rate that is fast. Making fast unneeded or needs that are random a web site just isn't good for anybody. As a result overloading of needs a website may decrease. In purchase to avoid this blunder, make your bot rest programmatically in between scraping processes. This might make scraper's bot look more individual towards the task that is anti-scraping. This will also not harm the web site. Washed the quantity that is tiniest of pages at a time simply by making requirements which are concurrent. Placing a timeout of around 10 to 20 moments and then carry on scraping. As mentioned earlier respect the robots.txt file.

Generally, scraper doesn't perform tasks which can be repetitive they browse through a website with random actions. But internet scraping bots will crawl within the pattern that is exact same they're programmed to do so. As mentioned earlier some websites have great mechanisms which can be anti-scraping. They are going to catch scraper's bot and can completely ban it. Now, how scraper can protect its bot from being caught? This is attained by Incorporating some clicks which can be random the page, mouse movements, and random actions which will make a spider appear to be a human.

Now, another problem is numerous sites change their layouts for all reasons and as a result scraper's scraper will don't bring information he/she are anticipating. With this, scraper ought to have a monitoring that is perfect that detects changes in their designs and then alert scraper because of the situation. Then these details can be used in scraper's scraper to function appropriately.

Websites show their content on the basis of which web browser scraper is using. Particular displays differently on different browsers. Let's just take the exemplory instance of Google search. The internet site may present "richer" content something more powerful and styled which might have hefty reliance on Javascript and CSS if the web browser (identified by the user agent) has advanced level capabilities.The problem with this particular is when doing any type of internet scraping, this content is rendered by the JS code and not the HTML that is raw the server delivers.

Bypass 504 HTTP Error

While scraping websites with splash it will give 504 HTTP Error so scraper will be able to bypass it by the following technique. So far when a scraper uses one splash instance to render javascript websites. He/she has to create a multiple splash instances with a load balancer in between them and then use them all at once to scrape or render javascript websites. This sounds cool right. So instead of having one splash instance or server a scraper can create three or more instances means a scraper can create a cluster of Splash instances and they all will collaborate between each other to render a specific web site.

Locality Sensitive Hashing

Now after successfully scraping the website successfully for searching the relevant literature scraper must have knowledge over Locality Sensitive Hashing. Locality-sensitive hashing (LSH) is definitely an strategy that is algorithmic hashes comparable input things in to the same "buckets" with a high probability.(how many buckets are much smaller than the universe of possible input products.) Since similar products result in the same buckets, this system may be used for data clustering and neighbor search that is nearest. It differs from traditional hashing techniques in that hash collisions are maximized, maybe not minimized. Instead, the strategy is seen in order to decrease the dimensionality of high-dimensional data; high-dimensional input things could be reduced to low-dimensional versions while preserving general distances between products. Hashing-based approximate neighbor search algorithms that are nearest generally use one of two main types of hashing methods: either data-independent methods, such as for instance locality-sensitive hashing (LSH); or data-dependent practices, such as Locality-preserving hashing (LPH).

Performing the phase of text similarity takes a simple machine-learning that is yet robust that will enable scrapers to effectively handle their productivity. That is of high value, enabling scientists to augment the process of the literary works search, with an aim towards focusing on more important objectives derived through the magazines faculties and taxonomy that is learning above. The machine learning arena is full of tools that address different degrees of individual maturity and comprehension of the domain [15].

A averagely complex device could be necessary to help the capacity to use various text mining tasks for the processes of mining and analysis of information for the relevant literary works search. Within our instance, Rapidminer had been the tool selected for the good and easy to use capabilities being graphical along with its out of the field text mining features. If the user is just a data mining beginner, KNIME could possibly be utilized also. The below shows a high-level view towards applying text mining. The above could then be scaled to effortlessly reproduce the appropriate literature search and summarization methodology proposed in literature while encouraging mass use through ease of use as a continuum to your web scraping workout. To show exactly the same, Figure 6a,b shows a text that is high-level procedure that is built to recognize term similarity between the papers being scraped.

Fig 6(a). Rapidminer process model deriving term similarity from the scraped literature search

Fig 6(b). text-mining sub-process components (tokenize, filter stop words, transform lower case and generate n-grams).

It could be seen that most regarding the components which can be below GUI-based elements that focus on increasing the simplicity of use as well as time and energy to value through the device.

V. CONCLUSION

So when to scrape websites and when to not and Now the very first thing a scraper have to check when scraping a website is terms of services and the robot.txt file now the robot.txt file is used by websites owners they can specify what website pages that can be scraped or not so if a scraper finds the web page that tries to scrape is forbidden in the robot.txt file In this case scraper is not allowed to scrape that website.

The second main thing a scraper have to check is does the website have a public API. If yes a scraper have to check if there are some limitations because some websites limit the amount of request. Next a scraper have to check if the API provides all the data it wants now in case of the API is paid it doesn't provide all the data and it has some limitations In this case the only solution you have is to use web scraping on it.

So reasons why scrapers generally use web scraping are data analysis and machine learning so data analysis is like evaluating the data use an analytical and statistical tools now of course in order to do some data analysis a scraper must have large amounts of data or what it is called data sets and the more data a scraper have the more accurate data analysis will be.

In web scraping the other big thing is machine learning now machine learning is like a set of algorithms that allows the computer or the software to be more accurate in predicting outcomes without being explicitly programmed and this also requires huge amounts of data now the more data scraper have the more scraper's system can learn by itself.

VI. FUTURE WORK

Now it comes to the maintenance and the stability of the spider. Now believe it or not the stability of the spider is like a couplet and 100 percent dependent on the website you are going to scrape and that's because if the website changes it's user interface. Scraper will end up with broken XPath expressions and CSS selctors and if scraper first created the spider they didn't use for example javascript to render the content and then after owners of website decided to redesign the website and make it fully dependent on javascript. So in this case scraper can end up with a broken spider from A to Z. so a scraper have to keep itself up to date with web scraping technology to avoid any kind of wastage and a desktop application scrapy is required which can scrape websites in the real time by using that desktop application.

REFERENCES

[1] BuzzSumo.com[Online]Available:https://buzzsumo.com/blog/filtering-the-worlds- content-5-ways-to-stay-ahead/. (Accessed: 28-Sep-2019).

[2] D. Pratiba, A. M.s., A. Dua, G. K. Shanbhag, N. Bhandari, and U. Singh, "Web Scraping And Data Acquisition Using Google Scholar," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), 2018.

[3] Mooney, S. J., Westreich, D. J., & El-Sayed, A. M. Epidemiology in the era of big data. Epidemiology, 26(3), 390. 2015.

[4] D. D. Prasetya, A. P. Wibawa, and T. Hirashima, "The performance of text similarity algorithms," International Journal of Advances in Intelligent Informatics, vol. 4, no. 1, p. 63, 2018..

[5] V. Bharanipriy and V. Prasad," WEB CONTENT MINING TOOLS: A COMPARATIVE STUDY," International Journal of Information Technology and Knowledge Management, Volume 4, No. 1, pp. 211-215. January-June 2011.

[6] Kushmerick Nicholas; Weld Daniel S.; Doorenbos Robert," Wrapper Induction for Information Extraction," Proceedings of the International Joint Conference on Artificial Intelligence, 1997.

[7] .Liu, L.; Pu, C.; Han, W.; , "XWRAP: an XML-enabled wrapper construction system for Web information sources," Data Engineering, 2000. Proceedings. 16th International Conference on , vol., no., pp.611- 621, 2000.7.I. Zoratti, "MYSQL Security Best Practices," 2006 IET Conference on Crime and Security, London, 2006, pp. 183-198.

[8] Tripathy, A.K.; Joshi, N.; Thomas, S.; Shetty, S.; Thomas, N., "VEDD-a visual wrapper for extraction of data using DOM tree," Communication, Information & Computing Technology (ICCICT), 2012 International Conference on , vol., no., pp.1,6, 19-20 Oct. 2012.

[9] S. M. Al-Ghuribi and S. Alshomrani, "A Comprehensive Survey on Web Content Extraction Algorithms and Techniques," 2013 International Conference on Information Science and Applications (ICISA), 2013.

[10] W. H.gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," International Journal of Computer Applications, vol. 68, no. 13, pp. 13–18, 2013.

[11] D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

[12] Glez-Peña, D.; Lourenço, A.; López-Fernández, H.; Reboiro-Jato, M.; Fdez-Riverola, F. Web scraping technologies in an API world. Brief Bioinform. 2013, 15, 788–797.

[13] Berglund, A.; Wg, X.S.L.; Boag, S.; Wg, X.S.L.; Chamberlin, D.; Wg, X.M.L.Q.; Almaden, I.B.M.; Fern, M.F.; Wg, X.M.L.Q.; Kay, M.; et al. XML Path Language (XPath) 2.0; W3C. 2010. Available online: https://www.w3.org/TR/xpath20/ (accessed on 5 December 2019).

[14] Haddaway, R.N. The Use of Web-scraping Software in Searching for Grey Literature. Grey J. 2015, 11, 186–190.

[15] 19. Dwivedi, S. Comprehensive Study of Data Analytics Tools. In Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 18–19 March 2016.