University of Technology, Jamaica
School of Computing and Information Technology

A comparative analysis of web crawlers for SCIT information retrieval

Prepared by

Genele Clarke-1700226

Ruth Elliott-1700229

Venus Farquharson-1700447

Britianna Morgan-1700236

Zidane Whittle-1700097

Presented to

Mrs. Sophia McNamarah

ABSTRACT

We aim to monitor the Performance and other significant findings of two Web Crawlers based on the data collected from five Devices along with research gathered from multiple empirical studies and peer reviewed journals and articles published in the past decade. With the results of this study, we seek to compare and review the significance of the different web Crawlers. This paper presents one Focused Crawler written in Java and one Distributed Crawler written in Python both programmed to fetch related links to the search query.

The experimental result of this research indicates that the distributed Crawler written in Python was considerably superior to the focus Crawler in terms of efficiency and accuracy for data collection.

Table of Content

Executive Summary

In 1958 the School of Computing and Information Technology(SCIT), then named the Department of Computer Studies, started at the then Jamaica Institute of Technology. Over the years SCIT students have participated in various projects to add to the technological advancement of the University and the country. Examples of these include Sapna, a SCIT born initiative where student members are able to expand their knowledge and participate in projects. In addition, SCIT has the honor of being among the first student branches in Jamaica of IEEE. Many of SCIT accomplishments are not properly documented or generally known. In recent times SCIT has considered international accreditation but however lacks the medium needed to keep track of the achievements of students and staff. As such this study aims to solve the problem of the University of Technology School of Computing and Information Technology not having a medium to sufficiently record students and staff achievements. By conducting this study we hope to improve student morale and boost motivation and attract potential students to the school.

Institutions providing higher education should have means of assuring students and prospective students that they are credible and providers of quality education. One of the most important measures of quality at a university that is used by accreditors in their assessment of a higher education institution is the records of its students' achievements. With the advent of the

internet many of the accomplishments are already in cyberspace. A web crawler is a useful tool to gather the information that is already available in cyberspace. Web crawlers allow for the automatic extraction of data from websites. For this research our objective is to explore the extent to which a web crawler can efficiently gather data on SCIT students' and staff current and past accomplishments. To achieve this objective our team created two web crawlers with the guidance of the iterative model software development model, a focused web crawler written in Java and a general purpose web crawler written in Python. For our study we used a comparative analysis that looks at the performance of two web crawlers across the 5 devices involved in the study. The web crawlers' effectiveness is measured in terms of their computer resource usage, their time complexity and their precision.The computer resource usage was monitored by observing the windows task manager of each device. We interpreted our findings by trying to find correlations and making comparisons and identifying data outliers. Our interpretations of the data will be reported by presenting the results followed by an explanation of our findings.

After collecting our data and carrying out our analysis it can be concluded that web crawler 2 written in python and is a general-purpose crawler is the more efficient crawler of the two. The precision of web crawler 2 was sixty-nine percent (69%) in comparison to web crawler 1 that has one percent (1%) precision. Web crawler two had the better precision. While observing the task manager across all five devices web crawler 1 remained consistent and close in range in all aspects of the task manager while web crawler 1 fluctuated and varied across far points on different machines. Thus making web crawler 2 more stable, faster, and more efficient than web crawler 1. Our recommendations at the end of this study is to run the crawler once a week, on a device with at least 4 Gb of RAM and 300 Gb of storage and for optimal results and utilization of the crawler we recommend that the crawler bot run on a device that has 8Gb of RAM and 500

Gb to 1 terabyte of storage. Also , a suitable seed url should be used such as google news, gleaner.com, observer.com or loop news.

CHAPTER 1

INTRODUCTION

In today's society organizations and people are striving for efficiency, reduction in cost, and high profitability of their business activities while saving time, (Samadi A., 2018). To achieve this feat, businesses use several strategies and tactics to be at the top of their business sector. Marketing and promotion is one of the most essential activities of any organization as it builds a customer base, increases sales and helps in decision making. The use of the internet and computers have rapidly increased due to more and more individuals adopting it as a part of their daily life routine, controlling their decisions and connecting across borders. Thus organizations have changed the way they market and promote themselves. Marketing is user-centered as it involves grouping individuals according to common features by understanding human patterns. (Derycke, Rouillard, Chevrin &Yves Bayart, 2020). Organizations develop these groupings through the use of internet marketing and as such they must understand the interaction between humans and computers.

In 1958 the School of Computing and Information Technology(SCIT), then named the Department of Computer Studies, started at the then Jamaica Institute of Technology. The Jamaica Institute of Technology later became the College of Arts, Science and Technology (CAST) in 1959. In 1986 CAST became the University of Technology, Jamaica. In 1998 SCIT was formed and later consolidated with the School of Engineering to form the Faculty of

Engineering and Computing, the second largest faculty in the University. During this time SCIT adopted a four(4) year university wide degree programme structure and a student-centred learning philosophy. Over the years SCIT students have participated in various projects to add to the technological advancement of the University and the country. Examples of these include Sapna, a SCIT born initiative where student members are able to expand their knowledge and participate in projects. In addition, SCIT has the honor of being among the first student branches in Jamaica of IEEE. Many of SCIT accomplishments are not properly documented or generally known. In recent times SCIT has considered international accreditation and as such having a platform to store and easily retrieve these accomplishments for display would increase the School’s visibility.

Statement of the Problem

SCIT students have participated in various projects and activities to showcase their skills and talents and increased their knowledge. However, the school has no way to sufficiently record the accomplishments students' have had over the years. Thus past and present students, stakeholders and the general public are not able to reflect on these achievements.

Purpose of the Study/Project

The purpose of this project is to develop an efficient web crawler that will allow for the retrieval of SCIT's accomplishments and achievements for stakeholders. This study seeks to establish an efficient way to automate the capturing of URLs of web pages related to information about exemplary achievements of SCIT present and past students and staff. This study will allow us to capture and record SCIT’s historical milestones to be stored in a central repository for ease of retrieval.

Significance

This project is important because it will improve student morale and boost motivation and attract potential students to the school. This will also increase the marketability of SCIT graduates as it informs the public of their accomplishments.

Research Questions

1. To what extent can a web crawler efficiently gather data on SCIT students’ current and past accomplishments?

Delimitations

- We will only use focus and general web crawlers to carry out this research because of financial constraints and because focused web crawlers are topic-specific and general webcrawler are page specific, it will allow us to collect web pages that are relevant to our area of interest.

Limitations

- There is a lack of previous research studies on the use of web crawlers to obtain information about academic performance.

Definition of terms

1. Web crawler - is a computer program/software that automatically searches the world wide web for specific topics that the users want.
2. Usability - the extent to which an user can use a product to achieve their goals with effectiveness, efficiency and satisfaction.
3. Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
4. Programming Language- can be defined as any language used for expressing a set of detailed instructions for a digital computer.
5. Scrapy framework- can be defined as an open source and collaborative framework for extracting the data you need from websites.It is mainly used in the python programming language.
6. Remote Method Invocation- The RMI (Remote Method Invocation) is an API that provides a mechanism to create distributed applications in java.

CHAPTER 2

LITERATURE REVIEW

Institutions providing higher education should have means of assuring students and prospective students that they are credible and providers of quality education. One such way this can be achieved is through gaining accreditation from the relevant accrediting agencies. According to Happe (2015) accreditation is the means by which it is ensured that an institution of higher education provides acceptable levels of quality and adheres to recognized standards. According to Accreditation Services for International School, Colleges and University(N.A) the standard of accreditation includes Student Welfare, Awards and Qualification and Marketing and Recruitment etc. One of the most important measures of quality at a university that is used by accreditors in their assessment of a higher education institution is the records of its students' achievements. According to Council for Higher Education Accreditation (2010) student achievement refers to the knowledge, skills and abilities that a student has attained as a result of engagement in a particular set of higher education experiences. Despite its importance many higher education institutions fail to keep adequate records of student achievements, (Council for Higher Education Accreditation, 2019). With the advent of the internet many of the accomplishments are already in cyberspace. A web crawler is a useful tool to gather the information that is already available in cyberspace. Web crawlers allow for the automatic extraction of data from websites. Such a tool can help in the collection of data that can provide a better picture of the quality of students an institution of higher education is producing.

Web Crawler

A lot of data is generated by the Internet and intranets. People typically have the option of getting the appropriate details from search engines. Thus, web crawlers are critical for the retrieval of information that crosses the Web as it downloads web documents that meet the needs of the user.

A web crawler is characterized as a program or software which navigates the World Wide Web and downloads web resources in a manner that is systematic, automated and methodical. (Kausar A. Dhaka V. Singh S. 2013)

The World Wide Web (WWW) is a data framework where documents and other resources are distinguished by Uniform Resource Locators (URL), which may be interlinked by hypertext, and are available over the Internet. Therefore, any search operations can be characterised as a process of traversing the linked structure of the Web. Web crawlers may take advantage of this to navigate the internet. The main aim of a web crawler is to retrieve Web pages and place them into a local repository. **(**Kausar A. Dhaka V. Singh S., 2013)

In this study, we will focus on the application of web crawler as it pertains to searching the internet and retrieving information.

Brief History

The first web crawler, called the "World Wide Web Wanderer" was created in 1993.It was initially used to calculate the size of the WWW, then subsequently used to track and download URLs which were then stored in the database of the world's first web search engine known as

Wandex (Mirtaheri S., 2014). Later came Aliweb, which was another early search engine that enabled users to upload the URL where the index of their website was manually written. These Indexes include a list of URLs along with keywords and explanations written by the site administrators. In 1994, The first "full text" crawler and search engine was launched, this was known as "WebCrawler". The “WebCrawler” allows users to browse the content of the website Instead of the site administrator’s provided explanations and keywords, this reduces the risk of conflicting results and enables a more efficient search proficiency. (Mirtaheri S. 2014)

While early crawlers dealt with comparatively limited volumes of data, modern crawlers, such as the one utilised by Google, need to manage a considerably larger volume of data, due to the rapid rise in the amount of the websites.

Process

Web crawling, a process of collecting web pages in an automated manner, is the primary and ubiquitous operation used by a large number of web systems and agents starting from a simple program for website backup to a major web search engine (N. Kumar, S. Awasthi, D. Tyagi, 2016).

A web crawler begins operation with an initial URL or set of URLs referred to as seed URLs (while focused crawlers utilize a topic description). The web crawler downloads the web pages from the seed URL and extracts new links from the pages that were downloaded. These web pages are indexed and then stored in a database. With the aid of these indexes the web pages can be later retrieved when required. The extracted URLs from these web pages are used to confirm that their related documents have already been downloaded. If the download was not completed, the URLs are once again allocated to web crawlers for further downloading. This process is repeated until there are no more URLs left to download. Therefore, web crawlers will recursively insert new URLs into the search engine's database repository. (Haas T., 2019)

Web Crawler Techniques

In general, web crawlers are classified into three types of crawling techniques: general purpose crawling, focused crawling and distributed crawling.

A general-purpose web crawler gathers as many web pages as possible from a given set of URLs and all their possible links. With this the crawler can fetch a large number of pages from various locations. The speed of the network bandwidth may be greatly reduced due to the large volumes of web pages being processed by the general-purpose web crawler. (Kausar A. Dhaka V. Singh S, 2013)

A focused crawler is programmed to search and retrieve documents with a specific topic, this will significantly reduce the number of downloads and network traffic. The purpose of the focused crawler is to specifically search for pages that are related to a set of predefined topics. It only covers the relevant areas of the web and which results in substantial savings in hardware and network resources. (Kausar A. Dhaka V. Singh S, 2013)

Last but not least, distributed crawlers. Several processes are used to crawl in distributed crawling, and sites for downloading from the Internet.

Distributed web crawling is a high performance computing technique whereby several computers are used by Internet search engines to index the Internet via web crawling. In order to crawl web pages, such systems can allow users to voluntarily give their own computing and bandwidth resources. Through spreading the load of these operations over many computers, costs that would otherwise be spent on operating huge computing clusters are avoided.

Programming Languages generally used to create crawlers

Web crawlers can be written in many different programming languages. However, when selecting a particular language; it is best to choose a language that meets the design requirements, needs of the user and, how well the programming language is suited to achieve the given tasks (Zhi-hang, T., & Jun, L. 2019). According to Liu, C., & Nie, N, (2020) the python language is best suited to create crawlers due to it being an object oriented and a high-level interpreted language. It also takes into consideration that the python language has a more easily understood syntax and the language can achieve accurate crawling and capturing of data more efficiently compared to other languages. The python language is equipped with many frameworks and extensions that are conducive to creating functional and efficient web crawlers. The python language is also more widely used based on reasoning that it is portable, scalable and it also has the ability to export and reuse functions that were built in other python projects due to its functions being able to operate independently (Xiang, L. C.,et al 2015). For instance, the scrapy framework and several library extensions such as the urlib, network library, re regular expression library and the lxml parsing library was utilized to create a crawler that was tasked with collecting information on particular clothing item (Zhi-hang, T., & Jun, L. 2019).

On the other hand, programming languages like ‘Go’ are used to create crawlers that are best suited to represent the user’s perspective. Go was used to create a universal web crawler called Kraaler which utilizes search engines such as chrome to gather data about parsing and the usage of HTTP. The Go language was selected due its unique features of being able to compile statistically linked binaries across multiple platforms (Panum, T. K et al, 2019). Developers also use languages such as Java to create simple but effective web crawlers. The java programming language is unique and mostly selected for its networking capabilities such as the java.net package and socket classes. It is also selected because of the way it makes use of the regular expressions. Java programming toolkits listed makes it very easy to retrieve URLs from the world wide web. It can also be noted developers use java because it allows them to use a feature called 'Remote Method Invocation', which is simply used to create distributed applications that can be executed from other java virtual machines. This technology allows the code to be platform independent like python (Peshave, M., & Dezhgosha, K. ,2005).

Challenges of Web Crawling

Another name for the process of web crawling is spidering (Society for Industrial and Applied Mathematics, 2019). Many genuine sites, specifically search engines, use spidering as a method for giving timely data. Web crawlers are used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code (Society for Industrial and Applied Mathematics, 2019).

Constant page updating can lead to numerous downloading of pages with each minute change in the structure of the page which might not be perceived by the user leading to wastage of bandwidth and time. Image processing happens with complex counts prompting delay in handling time and increases load on the network because of these unpredictable calculations.

failure of server may cause failure of whole framework because of nonattendance of circulated server system.

The web crawler can have uniformed structures, the web is a dynamic space that doesn't have a set norm for information arrangements and structures. Gathering information in an arrangement that can be perceived by machines can be a test because of the absence of consistency. For example, a site page can be made utilizing HTML, CSS, Java, PHP, or XML. The cycle of information extraction becomes challenging when the web crawlers need to organize information for a gigantic scope. The issue gets enhanced when the web crawlers need to separate information from a great many web sources relating to a particular outline. (quantzig, 2020).

Another challenge could be absence of context. Web creeping utilizes various procedures to download the substance that is pertinent to the user's query. The crawler centers around a specific subject, be that as it may, sometimes, the crawler in some cases may not be unable to discover relevant data. Subsequently, the crawler begins downloading countless unimportant pages. Therefore, developers need to discover strategies that focus on substance that intently looks like the search query.

Python is generally known as the best web scraper language. Due to the fact that it's more of an all-rounder and can deal with the vast majority of web crawling related activities easily. Beautiful Soup is perhaps the most generally utilized framework based on Python that makes crawlers using these languages take an easy route. Beautiful soup is a Python library that’s designed for fast and highly efficient web scrapers. One open source web crawler used by python is Scrapy which is a cross platform accessible.(Best programming languages for web scraping: Web crawling, 2021)

Although C and C++ offer incredible performance, the expense of building up a web crawling setup on these languages would be high. Henceforth, It isn't recommended to make a crawler utilizing C or C++ except if you are beginning an organization exclusively centered around web scraping. (Best programming languages for web scraping: Web crawling, 2021)

Efficiency

Efficiency refers to the capacity to reduce or eliminate wasted time, efforts, money, and/or resources when performing a task or accomplishing a goal. Like all software, crawlers also consume resources. Crawlers use network bandwidth to download webpages, RAM to support data structures and their algorithms, processing power to analyse and retrieve URLs, and disk storage to store URLs and the processed text that were fetched during the crawling process. The more complex the algorithm, the greater the consumption of these resources. (Srinivasan P. Menczer F. Pant G. 2004)

Based on our experience with both web crawlers and the research done into the relevant journal articles, we have decided on a basic set of measures that will allow for a comprehensive assessment of the efficiency and a fair comparison of both web crawlers. This includes utilizing the task manager application to track and monitor the CPU usage in percentage, the Memory Space in megabyte (MB), the disk storage in megabyte (MB), power usage and power usage trend, for fifteen (15) minutes over a period of five (5) days per web crawler. The results from these measures were then plotted on a bar graph in order to visually compare the two crawlers.

Precision

Precision can be referred to as the degree to which a process can replicate similar outcomes. In the case of web crawlers, this precision is based on the ratio of relevant records to irrelevant records. To calculate the precision, we divide the number of documents saved in the CVS by the total number of documents fetched by the crawler. (Wishard L. 2012)

Time Complexity

Another attribute that can be used to measure the efficiency of the crawlers is the time complexity. According to Krone, Ogden, & Sitaraman (2003) the time complexity is introduced in undergraduate data structures and algorithms courses to document the performance of algorithms, and to distinguish the efficiency of one algorithm from another. Time Complexity of an algorithm/program is not the measure of actual time taken for the program to be executed, rather it is the number of times each statement of the logic gets executed to produce the required output, Olivia & Barbosa (2016) . The time complexity provides a good objective and quantitative measure of the code of a web crawler and it is easy to relate to qualitative assessments, Batsakis, Petrakis, & Milios (2009). The time complexity of web crawlers will depend on the algorithms used to develop them and their particular use. In their study Menczer, Pant & Srinivasan (2004), concluded that the breadth first algorithm was more efficient than the best first algorithms when dealing with large amounts of links but the best first algorithm showed superior performance when crawling small amounts of links.

Algorithms used by Web Crawlers

According to Menczer, Pant & Srinivasan (2004) the breadth first algorithm crawls links in the order in which they are encountered. It starts at the base or seed URL which is usually a root domain and searches all neighbouring URLs that are found on the same level. After completing its first round of searches if the goal is not achieved the search will continue to the next level or subdomain and will continue in that order until its goal is reached (Singh A V, & Mishra A, 2014) If all domain levels are searched and the goal is not met it is reported as a failure. This type of algorithm is suitable for situations where the user is interested in gathering results from upper levels of a deeper tree using the shortest path possible. In order words it is best used when users know what they are searching for. The algorithm is simple, robust and offers high accuracy results as it searches and returns URLs related to a specific topic. Breadth first algorithm will not perform well on searches where all URLs are found on the same level and can lead back to each other. This will cause the algorithm to be in a constant loop. (Shukla V.&, Dharmendra R 2016). An example of this algorithm is a GPS navigation system such as Google Maps. This is how you are able to search for shops, stores and restaurants near you. This is also the algorithm used by social networks such as facebook to find persons close to you via schools you attended, address and many more.(Pavalam M, Kashmir R, Felix K and Jawahar M, 2011)

Depth first search algorithm also starts with a base/seed URL and traverses deeper through the child URL. If more than one child URL is found priority is given to the leftmost child which is then transverse to its child until no more child is found. It then returns to the unvisited child and another round of traversing continues in the same manner. This algorithm ensures that all URLs are collected. This is suitable if the user is interested in gathering all URLs of a specific website due to its nature of returning every URL. Depth first algorithm will not perform well if used to search for a specific topic as it would return URLs that are not related to the topic and also be in an infinite loop as it would never complete its goal.(Singh A V, & Mishra A, 2014). This algorithm can be used to create a sitemap of websites and the creation of puzzle games such as chess, mazes and sudoku.(Pavalam M, Kashmir R, Felix K and Jawahar M, 2011)

Another algorithm is called a Page Rank. According to Niechai (2021) this is an algorithm that ranks web pages by giving them a score of importance and authority. This algorithm has a valve called Page Rank and is assigned to pages which is used to measure the relevance of that page by counting the number of citations and backlinks to that page. This is done using the formula $PR(A) = (1-d) + d(PR(T1)/C(T1) + ...+ PR(Tn)/C(Tn))$ where PR(A) is the Page Rank of a given Page, d is the: Dumping Factor and T1 is the links. To determine the rank of a page A it is first recommended to find all the pages that link to A and the out links from A. the first page that link from A is represented by T1 which will give the number of outbound links to A the same is done for other pages link from A calling them T2, T3 and so on, the sum of values will be use to determine the rank if a webpage. (Singh A V, & Mishra A, 2014). This algorithm returns results of specific topics by traversing through pages searching for related URLs. An example of this can be seen with google searches as you go further on a search topic page results you will realise the results become less relevant to your search.This algorithm is suitable if the user wants to gather link that are already organized by relevance to their search topic with the first Page being the most relevant. (Singh A V, & Mishra A, 2014). This algorithm is primarily used by search engines such as google and bing in order to return search queries by users. .(Pavalam M, Kashmir R, Felix K and Jawahar M, 2011)

. The web crawler allows users to browse the web content of documents instead of the keywords and descriptors provided by the site administrators, reducing the risk of conflicting results and enabling a more efficient search proficiency. A Web crawler works by downloading the web pages and extracting new links from the pages that were downloaded. These web pages are indexed and then stored in a database. In general, web crawlers are classified into three types of crawling techniques: General Purpose Crawling, Focused crawling and Distributed Crawling. They also come with a few challenges. The constant page updating that it does can lead to numerous downloading of pages with each minute change in the structure of the page which might not be perceived by the user leading to wastage of bandwidth and time. The crawler also in some cases may not be unable to discover relevant data.

CHAPTER 3

METHODOLOGY

This chapter of the study serves to provide an explanation of the procedures undertaken to identify, select, process, and analyze the information that was gathered about the given research topic (Goundar.S, 2012). This methodology aims to provide a clear and concise explanation to fellow researchers and other interested parties about the research design and the steps taken to collect data. This is done to guide anyone who would like to replicate this study.

Research Design

This study uses a comparative analysis that looks at the performance of two web crawlers. It involved measuring the effectiveness of the two web crawlers which use the simple breadth first and pagerank algorithms. The web crawlers' effectiveness is measured in terms of their computer resource usage, their time complexity and their precision. The computer resources considered are memory, the cpu, disk and power usage.The crawlers were run on various computers for a set amount of time and the resource usage was recorded. These quantitative measurements were compared to determine the more efficient crawler. Each member of the research team possesses machines that are unique in specification to each other and as such each member executes the crawler programs to identify how the program would behave on various machines. The utilization of a comparative analysis allowed the researchers to make effective comparisons of the data collected from all devices involved in this study.

System Development Process

For this project two web crawlers were created, a focused web crawler and a general purpose crawler. The focused web crawler was created in Java and it used the simple breadth first algorithm. According to Menczer, Pant & Srinivasan (2004) the breadth first algorithm crawls links in the order in which they are encountered. As such it will crawl all the links on the starting page and select one of those links and continue crawling. This web crawler's seed url was set to:
https://jamaica.loopnews.com/content/utech-students-win-500000-sagicor-innovation-challenge and the key words were "utech" and "universityoftechnology". The general purpose web crawler was created in Python and it used the pagerank algorithm. According to Niechai (2021) the pagerank is an algorithm that ranks web pages by giving them a score of importance and authority. The seed url for this crawler was set to: https://www.google.com/search?

For the development of these crawlers the software development life cycle was used. The software development life cycle (SDLC) is the detailed process undertaken to develop and deploy a software (Okesola, O. J, et al. 2020, p. 26). The SDLC process can be broken down into six phases, the planning and requirement analysis phase, defining requirements, design, build, testing and deployment and maintenance phase. There are several SDLC models such as Waterfall, Spiral, Iterative and many more.The Iterative SDLC model approach will be used to build the web crawlers to be used for this project. The iterative SDLC model splits the project requirements into smaller portions which are then designed, built, tested and implemented. After every implementation the code is reviewed and the process of designing, developing and testing is repeated until a fully functioning web crawler is deployed (Okesola, O. J, et al. 2020, p. 28). The iterative model lowers the potential risk of failure, it is faster at providing functional deliveries compared to some SDLC models, due to the constant reviewing of the code helps to improve the quality of the finished product and the iterative model are best suitable for projects that utilizes new technology, the model allows the user to have a better understanding the technology (Verma, S. 2014, p. 109).

Evaluation of the Web Crawlers

To evaluate the efficiency of the web crawlers the researchers collected data on the use of system resources such as memory, the cpu, disk.and power usage by the web crawlers across all devices involved in the study by monitoring the windows task manager. The researchers also calculated the time complexity of both web crawlers in order to estimate how much time they will take to execute statements of codes in their algorithms regardless of which device the web crawlers run on. For this project, the researchers also calculated the estimated precision of each web crawler by dividing the number of relevant links retrieved by the crawlers by the total number of links retrieved.

Data Analysis

Data collected from the team observation of the web crawlers will be examined and organized to address the initial proposed research questions. A constant comparison analysis will be conducted on the data. The purpose of this analytical process is to continuously compare the interpretation and findings with current data as it is acquired throughout the data collection process(Onwuegbuzie, A. J., Dickinson, W. B., Leech, N. L., & Zoran, A. G., 2009). For our analysis we also found the maximum, minimum and numerical averages of the data we collected from the task manager of each device. We interpreted our findings by trying to find correlations and making comparisons and identifying data outliers. Our interpretations of the data will be reported by presenting the results followed by an explanation of our findings.

This analysis technique can be used to process many data types. For our study the data was collected in two formats, the team's notes and memory from the live sessions along with the notes made from the recorded sessions that will be recorded into an excel spreadsheet. The data collected was labeled, similar labels were then grouped together after which the researchers were able to match notes from both methods and provide a summary of the findings from each grouped data. Data collected was grouped according to features of the task manager and compared and summarized to be displayed as charts and tables.

Procedures

For this project an observational technique was conducted separately by each member of the research team on their computer. Each group member has a copy of the web crawlers and downloaded an appropriate IDE to successfully run each crawler. The team dedicated one week to collect data on the crawlers from the task manager. During this week each member ran the crawlers for a maximum of 15 minutes each for 5 days. While running the crawlers the members used a third party or the built in screen-recorder to capture the task manager. At the end of the 15 minutes the recording was stopped and rewatched by the member and they made note of the highest and minimum values for memory, the cpu, disk and power usage.

To calculate the time complexity we counted the number of operations performed by the codes of the web crawlers. This also involved us analyzing the codes of the web crawlers line by line and calculating the Big O of each operation and using the highest order term to represent the Big O of the algorithm.

The research team will also calculate the precision which is the number of relevant links retrieved by the crawlers divided by the total number of links retrieved. For the focused web crawler we set the crawler to crawl 1000 links, as this is the exact amount it can go through in 5 minutes and use the links saved to the csv file as the number of relevant links retrieved. As such the formula to be used to calculate the precision for this crawler will be:

Precision = (total number of links saved to the csv file / 1000)

For the general crawler we calculated the precision by dividing the total number of links saved to the csv file by the total number of search results crawled by the web crawler. As such the formula to be used to calculate the precision for this crawler will be:

Precision = (total number of links saved to the csv file / total number of search results crawled by the web crawler)

CHAPTER 4

RESULTS AND FINDINGS

Introduction

This chapter of the study serves to provide a detailed analysis of data collected in an effort to answer the research topic. Data was collected by the research team. Each team member utilized their machines to run two web crawlers for fifteen(15) minutes each day for five (5) consecutive days. The task manager was then observed using a third-party screen recorder to capture the CPU usage, memory usage, disk usage, performance, and performance usage. In which the highest and lowest results were recorded. Due to each team member possessing different computers the specifics of each were also recorded. Each device was given a code name for simplicity during analysis. After data collection was completed, the averages of the five(5) days were calculated and grouped for analysis. The findings were used to highlight trends and patterns. The data presented showcases the extent to which a web crawler can efficiently gather data on SCIT students' current and past accomplishments.

Analysis of two different web crawlers efficiency for information retrieval

Device Specifications	
Machine Specifications	Device Code
IdeaPad S145-15IWL/ intel(R) Pentium(R) CPU 5405U @ 2.30H+GHz /4.00GB RAM/ 64-bit OS, x64-based processor	Device 1
Intel Core i3-4030U CPU1.90GHz RAM 6GB/	Device 2
Intel(R) Core(TM) i5-7200U CPU @ 2.50GHZ 8.00 GB RAM 64 bit operating system	Device 3
LAPTOP-6LP3T2IF, Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz, 64-bit operating system	Device 4
IdeaPad 3 1511L05 / Intel(R) / Core(TM) i3-1005G1 CPU @ 1.20GHz / 8.00 GB 1.19 GHz 64-bit OS, x64-based processor	Device 5

Table 1: Table showing each researcher specification and code name to be used during discussion and analysis

Web crawler Dictionary			
Name	Type	Language	Code Name
Java Web Crawler	Focused	Java	Webcrawler1
Python Crawler	General-purpose	Python	Webcrawler2

Table 2: Table showing basic information on each crawler used in this study and their code name used as a reference throughout the project.

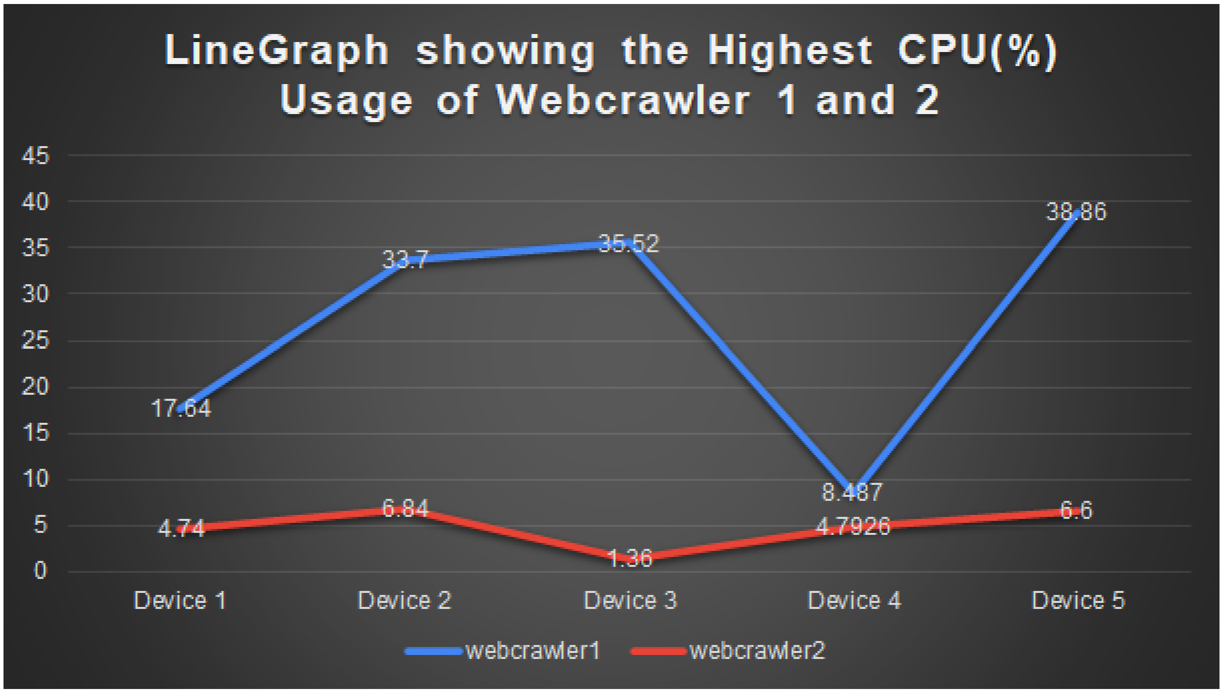


Figure 1: Line graph showing the CPU usage of the two web crawlers.

Based on the graph we can conclude that webcrawler2 takes less processing to complete its task. Across all five(5) devices webcrawler2 shares values in close range of each other. We can then assume that based on CPU usage webcrawler2 uses less process and time to complete its task making it an ideal web crawler for information retrieval. However, for webcrawler1 computer that possesses an intel core i58250U processing unit may experience significantly less processing time, otherwise, computers with other processing units maybe notice high consumption of CPU usage which could result in slow down of other processes if a user chooses to run many applications concurrently. According to Bozhkor, 2017 the intel core i58250U offers a bigger performance than its predecessors due to its hyper-threading technology.

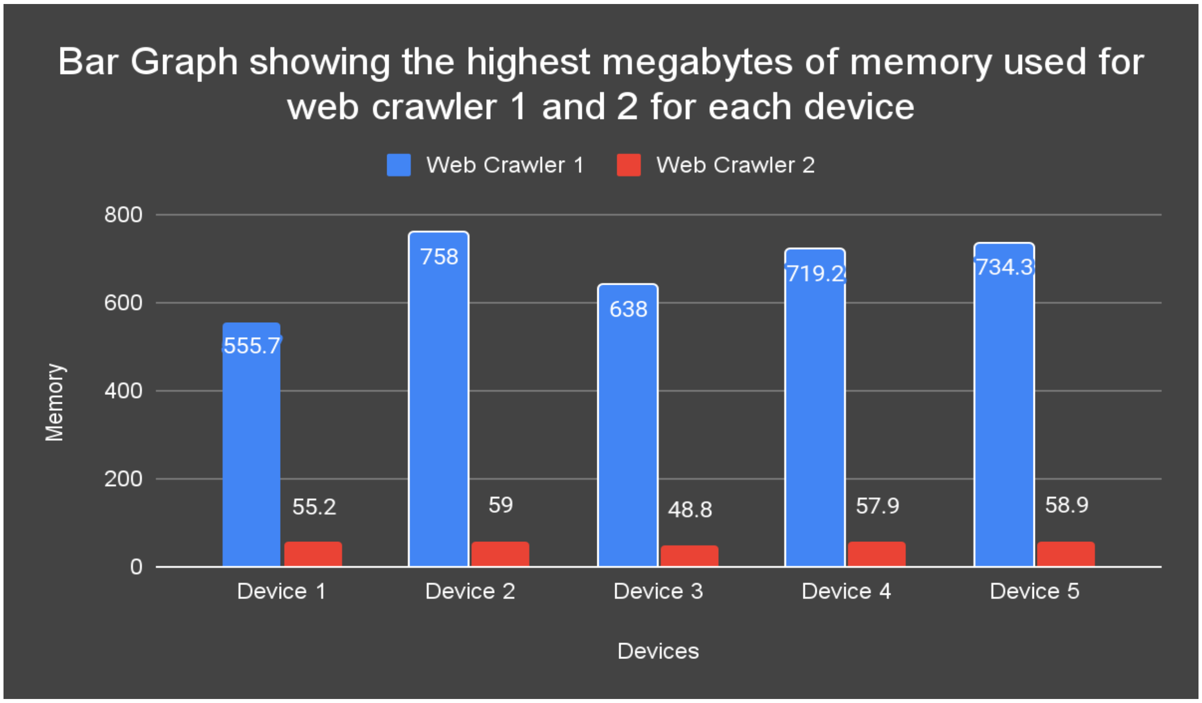


Figure 2: Bar Graph showing the highest memory consumption of the crawlers on the different devices.

As observed in the bar graph above, web crawler 1 consumes more megabytes of memory across all five devices compared to web crawler 2 which consumes drastically less. Crawler 1 shows a trend of memory consumption ranging from 555.7 upwards to levels as high as 758 megabytes. On the other hand, web crawler 2 shows a gradual increase from 48.8 and peaks at 59 megabytes of memory usage.

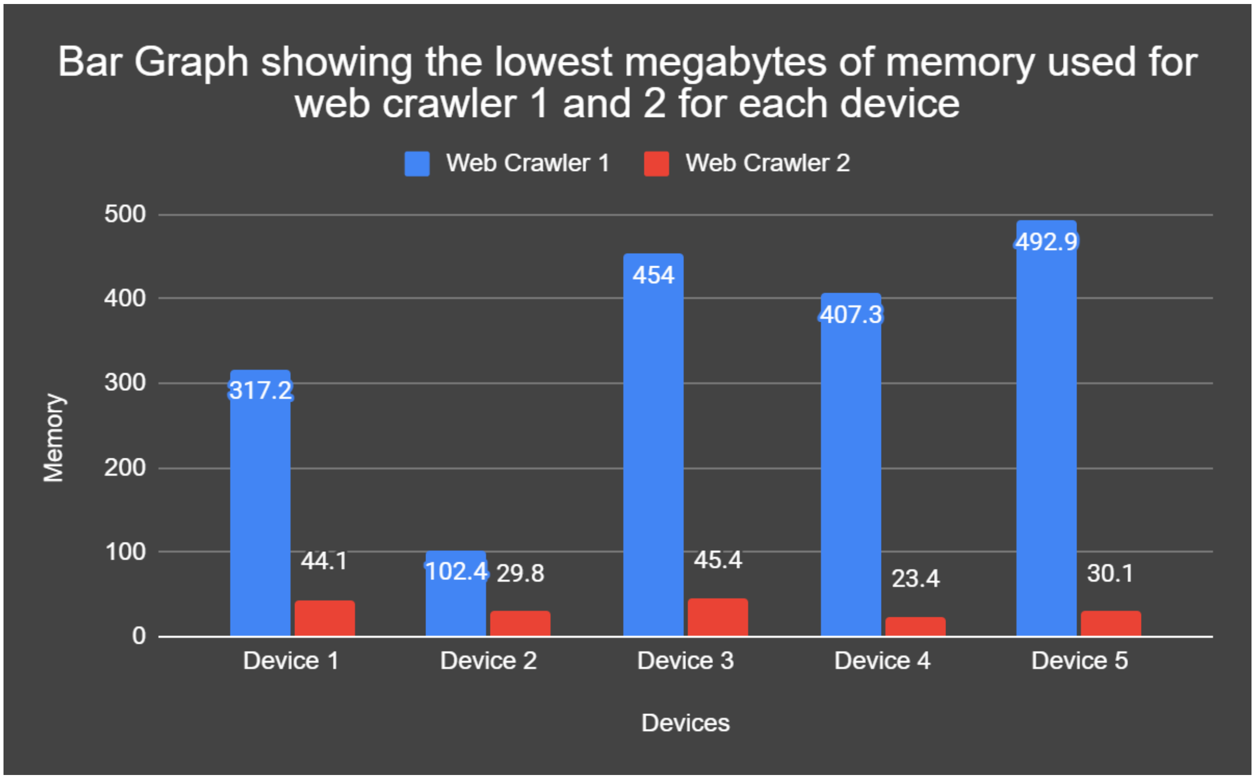


Figure 3: Bar Graph showing the lowest memory consumption of the crawlers on the different devices

As observed in the bar graph above, web crawler 1 consumes more megabytes of memory across all five devices compared to web crawler 2 which consumes drastically less. Crawler 1 shows a trend of memory consumption ranging from 492.9 downwards to levels as low as 102.4 megabytes. On the other hand, web crawler 2 shows a gradual decrease from 45.4 and drops to 23.3 megabytes of memory usage.

Based on figures 4 and 5. It can be said that web crawler 2 is more suitable to use due to it having drastically less impact on all five devices memory usage compared to crawler 1.

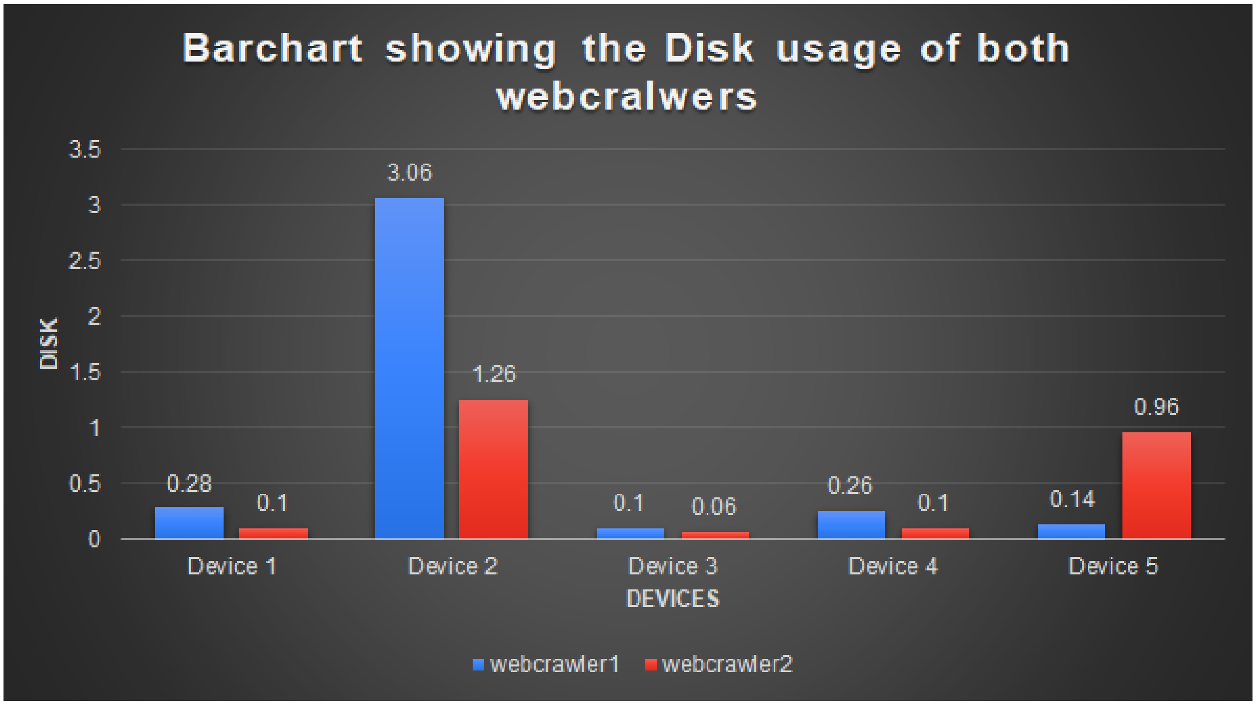


Figure 4: Barchart showing the highest disk space usage for each crawler on the different devices.

Based on the chart above the disk usage between the two crawlers is close together. The disk tells us how many resources a program is consuming whether they are reading or writing the disk. However, we can see a major cap on device 2 between the two devices. Device 2 is the only device among the five that has 6GB RAM and maybe the cause of such a high number. According to Laptop PC made simple, a 6GB RAM PC possesses 2 chips one of 2GB and 4GB, therefore, making it one the most unpredictable and unstable RAM size chips for PCs since it can either behave as a 2GB or a 4GB RAM at random.

Webcrawler1	Power Usage	Power Usage Trend
Device 1	Low	Very Low
Device 2	Very Low	Very Low
Device 3	Very Low	Very Low
Device 4	Very Low	Very Low
Device 5	Very Low	Very Low

Table 3: Table showing the lowest power usage and trend of webcrawler1 on average across all five(5) devices

Webcrawler2	Power Usage	Power Usage Trend
Device 1	Very Low	Very Low
Device 2	Very Low	Very Low
Device 3	Very Low	Very Low
Device 4	Very Low	Very Low
Device 5	Very Low	Very Low

Table 4: Table showing the lowest power usage and trend of webcrawler2 on average across all five(5) devices

Based on table 3 and 4 we can see where there is a similarity in power usage between both crawlers. They both remained very low across all devices therefore both web crawlers can successfully run on low battery status.

Webcrawler1	Power Usage	Power Usage Trend
Device 1	High	Low
Device 2	Very High	Low
Device 3	Very High	Low
Device 4	High	Very Low
Device 5	Very High	Moderate

Table 5: Table showing the highest power usage and trend of webcrawler1 on average across all five(5) devices

Webcrawler2	Power Usage	Power Usage Trend
Device 1	Very Low	Very Low
Device 2	Low	Very Low
Device 3	Very Low	Very Low
Device 4	Low	Very Low
Device 5	Very Low	Very Low

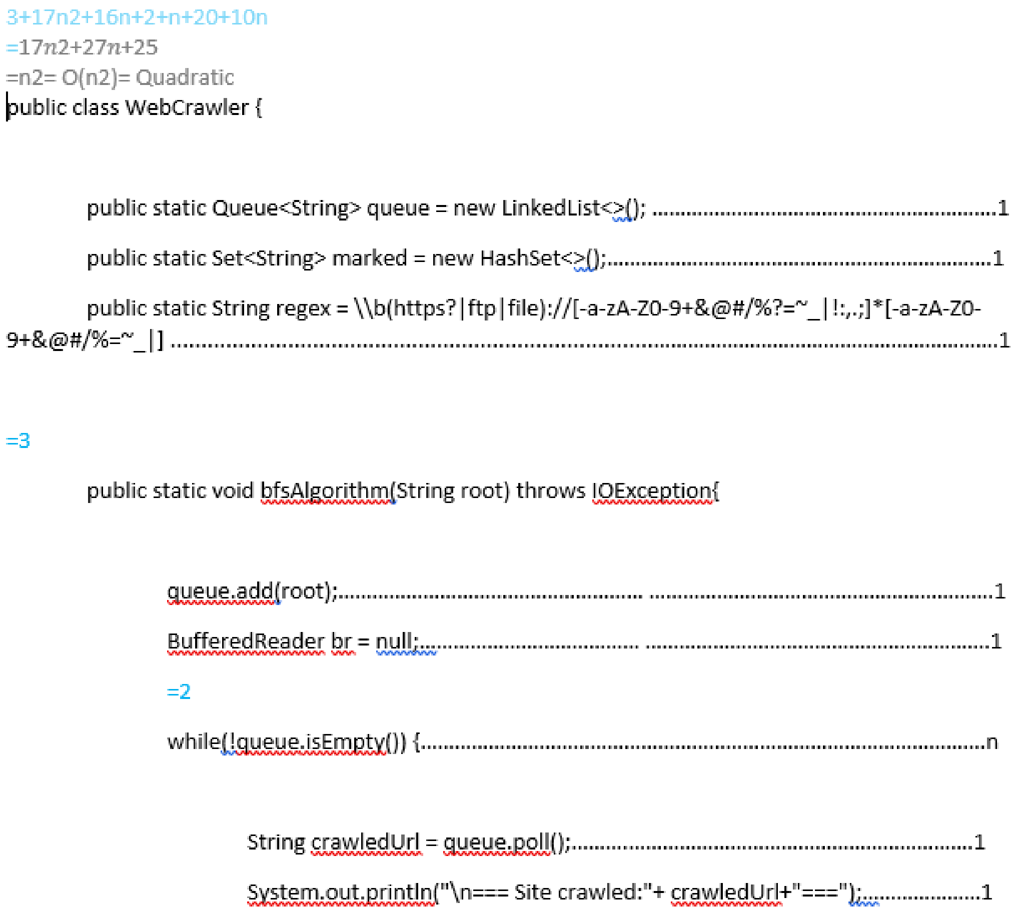
Table 6: Table showing the highest power usage and trend of webcrawler2 on average across all five(5) devices

Based on table 5 and 6 we can state that webcrawler2 remained more consistent between the two as well as maintained a low power usage while webcrawler1 fluctuate between high and very high. We can therefore conclude that webcrawler2 is more likely to successfully run on any machine no matter the power limit. While webcrawler1 consumes a lot of power and may not be able to run on a low battery efficiently.

Time complexity

Time complexity is referred to as the computational hardness or difficulty that describes the amount of time the computer takes to run an algorithm. It measures the time that is taken to execute each line or statement of code in an algorithm. Developers can use time complexity to assess whether the program is efficient or whether we need to use a different approach that takes less time.

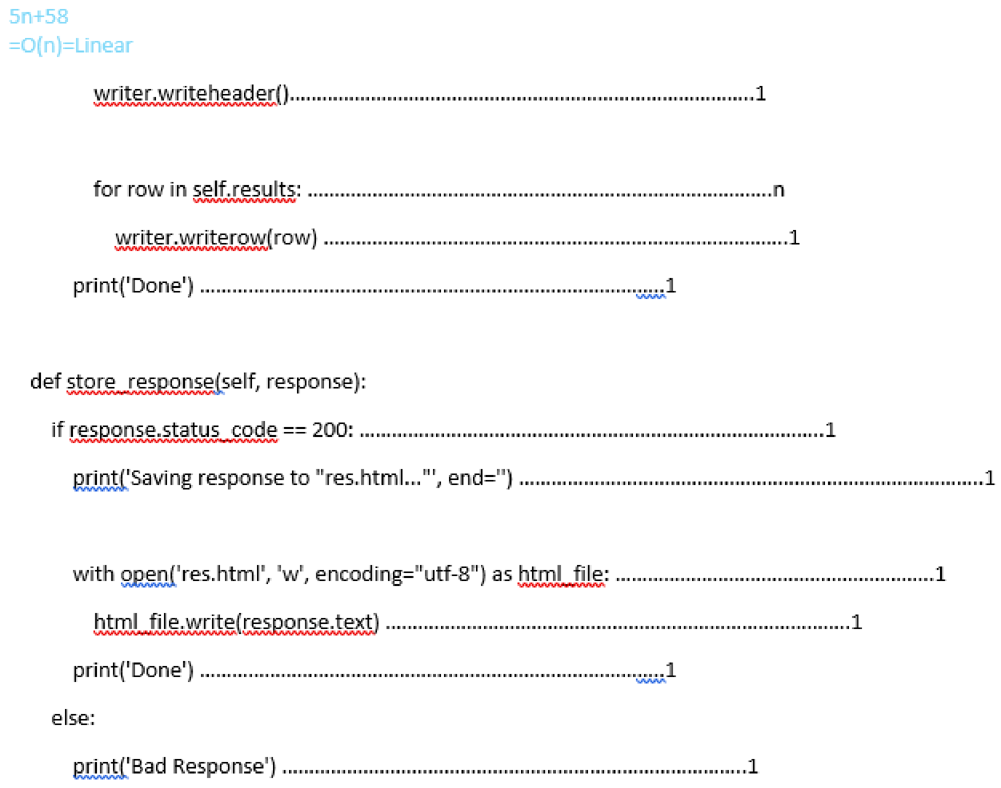
Webcrawler1



```
3+17n2+16n+2+n+20+10n
=17n2+27n+25
=n2= O(n2)= Quadratic
public class WebCrawler {


          public static Queue<String> queue = new LinkedList<>(); ..........................................1

          public static Set<String> marked = new HashSet<>();..............................................1

          public static String regex = \\b(https?|ftp|file)://[-a-zA-Z0-9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-
9+&@#/%=~_|] ...........................................................................................1


=3
          public static void bfsAlgorithm(String root) throws IOException{


                    queue.add(root);.......................................  ...................................1

                    BufferedReader br = null;.................................  ..................................1

                    =2

                    while(!queue.isEmpty()) {.....................................................................n


                              String crawledUrl = queue.poll();.................................................1

                              System.out.println("\n=== Site crawled:"+ crawledUrl+"===");....................1
```

The java program ran in quadratic time which means it ran O(n2). Due to the fact that this code had a nested while loop.

Webcrawler2



5n+58
=O(n)=Linear

```
            writer.writeheader()......................................................................................1


            for row in self.results: ......................................................................................n

                writer.writerow(row) ......................................................................................1

        print('Done') ......................................................................................1


def store_response(self, response):

    if response.status_code == 200: ......................................................................................1

        print('Saving response to "res.html..."', end='') ......................................................................................1


        with open('res.html', 'w', encoding="utf-8") as html_file: ............................................................1

            html_file.write(response.text) ......................................................................................1

        print('Done') ......................................................................................1

    else:

        print('Bad Response') ......................................................................................1
```

The python program ran in linear time which means it ran in O(n). this was because of the for loops within the code.

Line Graph Showing Program time for each crawler

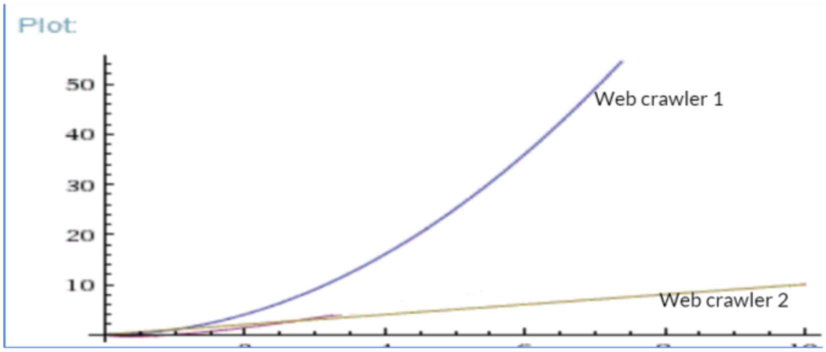


Figure 6: A graphical representation showing the differences of each web crawler program's time.

Linear Time Complexity refers to the complexity of an algorithm or program that grows in direct proportion to the magnitude of the input data. As a general guideline, you should strive to keep your functions running below or within this time-complexity range, but this isn't always practicable.

When the execution time of an algorithm increases by n2 with the length of the input, it is said to have a non-linear time complexity. In general, nested loops fall into the O(n)*O(n) = O(n2) time complexity order, where one loop takes O(n) and if the function comprises loops within loops, it takes O(n)*O(n) = O(n2).

Big O complexity for constant, linear, and quadratic log

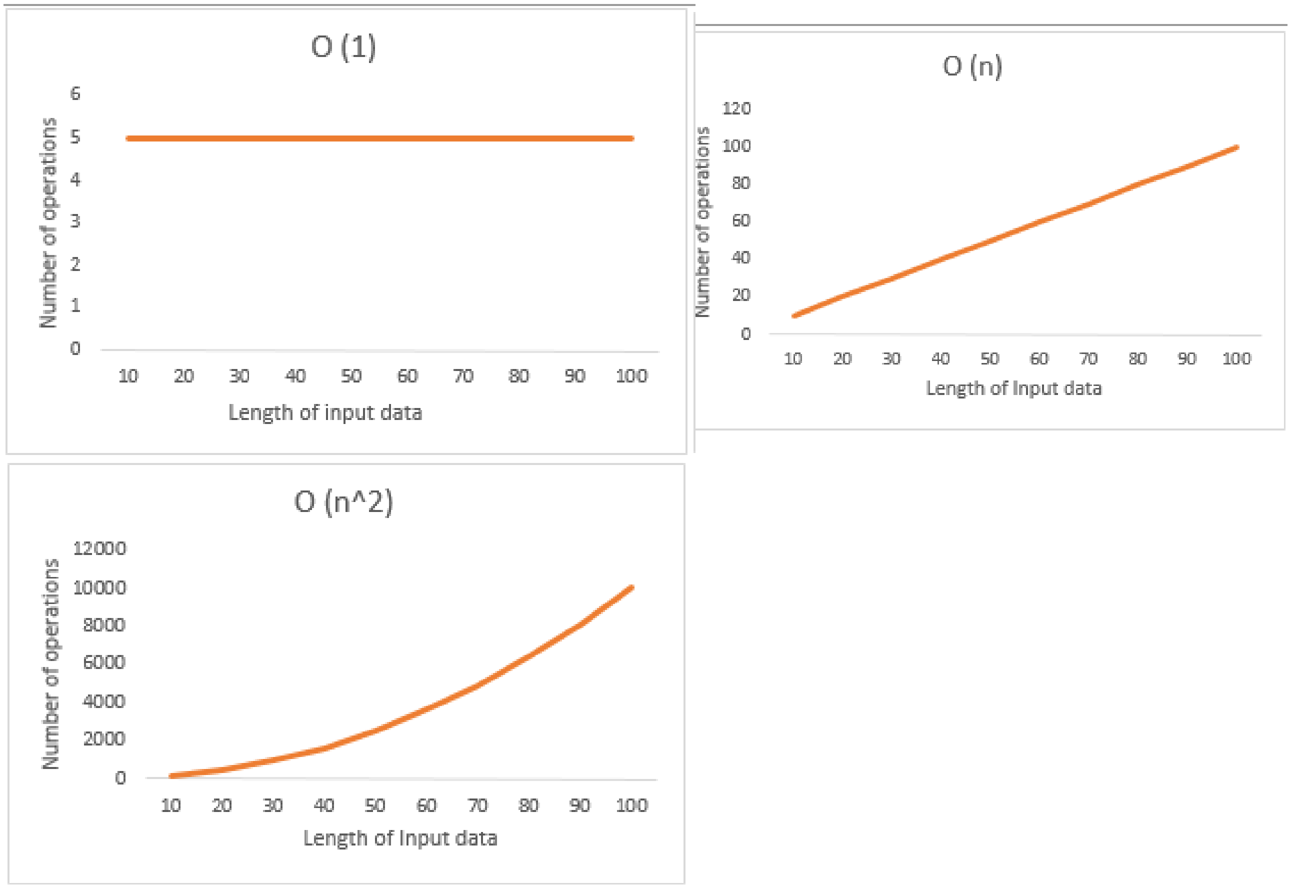


Figure 7: Showing the Big O complexity for constant, linear, and quadratic log.

As shown in the graph O(1) which represents an algorithm or program running in constant time. This takes up the least amount of time to be executed by the computer. As shown, no matter the length of the data input the number of operations always remains the same. As for O(n) which represents a program or algorithm running in linear time as the length of data increases the number of operands also slowly increases at a very slow pace while as for O(n2) for each time the data is increased a drastic amount of operations are performed. This requires a significantly more amount of time to be executed unlike the linear or constant algorithms.

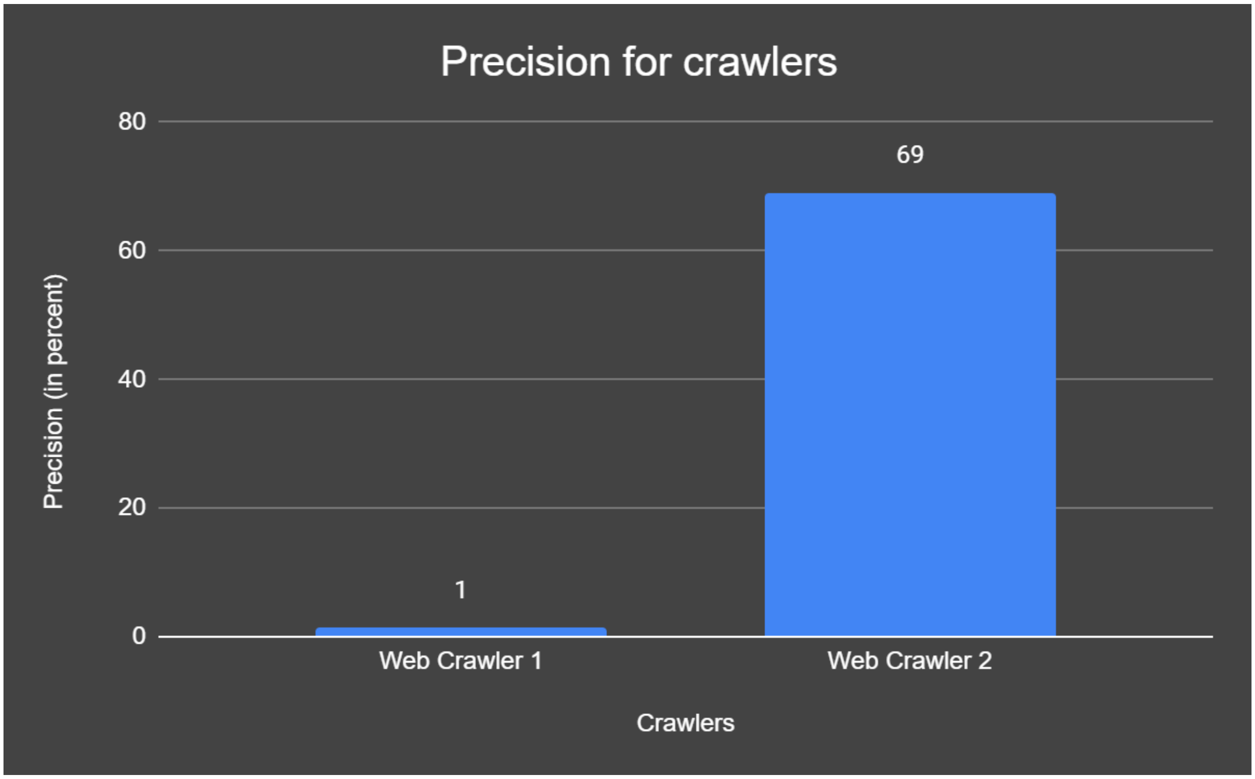


Figure 8: Bar graph showing the precision of each web crawler.

According to the diagram above Web Crawler 1 has a precision of 1% while Web crawler 2 has a precision of 69%. This means that while completing a crawling task Web Crawler 1 spends 99% of its effort going through irrelevant items while Web Crawler 2 only spends 21% of its effort going through irrelevant links. As a result it can be concluded that Web Crawler 2 is more effective at retrieving relevant results about the students of the school of computing and information technology than Web Crawler 1. The disparity in the precision of the crawlers can be attributed to the domains in which the crawlers run. Web Crawler 2 stays within Google's search results while Web Crawler 1 has no restriction as to which domain it operates. Lu, Zhan, Zhou, & Dengchao (2016) concluded that crawlers are able to yield good precision by limiting the domains in which they crawl and this was the case with Web Crawler 2. Furthermore, Gasparetti, Micarelli, & Sansonetti (2014) concluded that when a crawler is able to crawl a high percentage of relevant pages and reduce the amount of irrelevant pages it traverses, it will be able to achieve significant savings in the usage of system resources as was the case with our experiment. Web Crawler 2 traversed less irrelevant pages and as such it utilized less of the system resources on all devices.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

Conclusion

Over the years SCIT students have achieved various accomplishments through participation in various activities and projects. However, the school has no way to sufficiently display these achievements. This study sought to establish a unique and efficient way of easily sharing information with past, present, and potential students and stakeholders of SCIT in hopes to improve student morale and boost motivation, attract potential students to the school and increase graduate's marketability. The research question is:

- To what extent can a web crawler efficiently gather data on SCIT students' current and past accomplishments?

The researchers opted to use their machines due to the coronavirus to observe the behavior of two web crawlers to be compared in order to find which of two is more efficient and recommend its usage for information retrieval for the SCIT department.

To what extent can a web crawler efficiently gather data on SCIT students' current and past accomplishments?

After conducting the study based on the research question to ascertain the web crawler efficiency to gather data on SCIT students' current and past accomplishments, data was data collected by each member of the research team and analyze, it can therefore be concluded that web crawler 2 written in python and can be considered as a general-purpose crawler is the more efficient crawler of the two. Efficiency is the ratio of a program output to input. It is a basic measure that can be used to benchmark programs against each other. In order to differentiate the efficiency of the web crawler, the two were constantly compared using common elements. The precision of web crawler 2 was sixty-nine percent (69%) in comparison to web crawler 1 that has one percent (1%) precision. Precision has two main elements, the relevant links and the total number of links gathered. Both crawlers gathered and stored links in a text and HTML file after which the relevant links were converted to a CSV file. The web crawlers were designed to be as accurate as possible by gathering only the relevant documents. Therefore the higher the precision the more accurate and up-to-date the web crawler is. Hence for web crawler one to have a precision of 1% is poor making for web crawler two that have a precision of 69% more effective. When comparing the time taken for each crawler to execute, crawler 2 took less time than crawler 1 expressed as O(n) which is linear. While observing the task manager across all five devices web crawler 1 remained consistent and close in range in all aspects of the task manager while web crawler 1 fluctuated and varied across far points on different machines. Thus making web crawler 2 more stable, faster, and more efficient than web crawler 1.

Recommendations for efficient web crawler information gathering of SCIT data

In order to successfully and efficiently automate the retrieval of information based on Scit related areas such as achievements of students. It is also recommended that the crawler be run once a week, on a device with at least 4 Gb of ram and 300 Gb of memory. This process will be used to check for updates of relatable links. However, for optimal results and utilization of the crawler; it is recommended that the bot run on a device that has 8gb of ram and 500Gb to 1 Tb of memory. It is also recommended by the researchers that a suitable seed url be used such as google news, gleaner.com, observer.com or loop news. A suitable seed url will result in more potential articles about Scit being recommended and crawled..

References

Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, *12*(1), 106.

Batsakis, S., Petrakis, E. G. M., & Milios, E. (2009). Improving the performance of focused web crawlers. Data & Knowledge Engineering, 68(10), 1001–1013. doi:10.1016/j.datak.2009.04.002

Beggs, J. M., Bantham, J. H., & Taylor, S. (2008). Distinguishing the factors influencing college students' choice of major. College Student Journal, 42(2), 381-395.

Cameron, J. (2005) 'Focussing on the Focus Group', in Iain Hay (ed.), Qualitative Research Methods in Human Geography, 2nd ed., Oxford University Press, Melbourne, Chapter 8.

Creswell , John (n.d.) *Mixed-Method Research: Introduction and Application* University of Nebraska Lincoln Chapter 18

Council for Higher Education Accreditation. (2010). EFFECTIVE PRACTICES: THE ROLE OF ACCREDITATION IN STUDENT ACHIEVEMENT*. Washington. Retrieved from https://www.chea.org/sites/default/files/other-content/Effective%20Practice%20Revised3%20UPDATED.pdf

Denis Shestakov (July, 2019). Current Challenges in Web Crawling

The Issues and Challenges with Web Crawler: Web Crawling. (2020, September 14). Retrieved January 11, 2021, from https://www.quantzig.com/blog/web-crawler-challenges-crawling

Derycke A., Rouillard J, Chevrin V. &Yves Bayart (2020, January 16) When Marketing meets HCI: multi-channel customer relationships and multi-modality in the personalization perspective retrieved from https://hal.archives-ouvertes.fr/hal-02442165/document

Dix A, Finlay J., Abowd G.D, Beale R (2004) *Human Computer Interaction Book*(3rd ed) Essex, England Pearson Education Limited

Dr. Naresh Kumar, Shivank Awasthi, Devvrat Tyagi (December, 2016). Web Crawler Challenges and Their Solutions, International Journal of Scientific & Engineering Research, Volume 7, Issue 12

Eberle, D., Berens, G., & Li, T. (2013). The impact of interactive corporate social responsibility communication on corporate reputation. *Journal of business ethics*, *118*(4), 731-746.

Few, S., & Edge, P. (2007). Data visualization: past, present, and future. *IBM Cognos Innovation Center*.

Figueiras, A. (2014). How to Tell Stories Using Visualization. *18th International Conference on Information Visualisation*, 18-26.

Gershon, N., & Page, W. (2001). What Storytelling Can Do for Information Visualization. *COMMUNICATIONS OF THE ACM Vol. 44, No. 8*, 31-38.

Goundar, Sam. (2012). Chapter 3 - Research Methodology and Research Method.

Graham Edward (2020) Showcasing Student Work Retrieved from https://www.nea.org/professional-excellence/student-engagement/tools-tips/showcasing-student-work

Haiying Long (2014) An Empirical Review of Research Methodologies and Methods in Creativity Studies (2003–2012), Creativity Research Journal, 26:4, 427-438, DOI: 10.1080/10400419.2014.961781

Happe, E. H. (2015, November 12). *The Importance of Accreditation*. Paralegal Blog.

https://www.paralegal.edu/blog/the-importance-of-accreditation

Lee, B., Riche, N. H., Isenberg, P., & Carpendale, S. (2015). More Than Telling a Story: Transforming Data into Visually Shared Stories. *Visualization Viewpoints*, 84-90.

Liu, C., & Nie, N. Film Comment Collection Technology and Realization of Distributed Web Crawler Based on Python.

N.d (2019, December 12) The mathematics of prey detection in spider orb-webs. Retrieved from https://www.sciencedaily.com/releases/2019/12/191212104054.htm

Madaus G.F., Stufflebeam D., Scriven M.S. (1983) Program Evaluation. In: Evaluation Models. Evaluation in Education and Human Services, vol 6. Springer, Dordrecht. https://doi.org/10.1007/978-94-009-6669-7_1

Majid, Umair. (2018). Research Fundamentals: Study Design, Population, and Sample Size. Undergraduate Research in Natural and Clinical Science and Technology (URNCST) Journal. 2. 10.26685/urncst.16

Menczer, F., Pant, G., & Srinivasan, P. (2004). Topical web crawlers. ACM Transactions on Internet Technology, 4(4), 378–419. doi:10.1145/1031114.1031117

Srinivasan P. Menczer F. Pant G. (2004). A General Evaluation Framework for Topical Crawlers. Retrieve from https://carl.cs.indiana.edu/fil/Papers/crawl_framework.pdf

Minimum P. (2018, May 18) Human-computer interaction and digital advertising retrieved from https://martechtoday.com/human-computer-interaction-and-digital-advertising-215884

.Murphy A (2019, June 12) 7 Reasons to Study Human-Computer Interaction retrieved from https://www.academiccourses.com/article/seven-reasons-to-study-human-computer-interaction/

MYERS, B., HOLLAN, J., & CRUZ , I. (1996). Strategic Directions in Human-Computer Interaction. *ACM Computing Surveys*, 795-808.

N.d (n.d) About the Faculty retrieved from https://www.utech.edu.jm/academics/colleges-faculties/fenc

N.d (n.d) *Standards for Accreditation and Key Performance Indicators International Universities* https://www.asicuk.com/documents/ASIC-Standards-for-Accreditation-Int-Universities.pdf

Niechai, V. (2021, February 3). *Google's PageRank Algorithm: Explained and Tested*.

Link-Assistant.Com. https://www.link-assistant.com/news/google-page-rank-2019.html

Nicastro D. (2019, February 6) How Human-Computer Interaction Can Help Marketers retrieved from https://www.cmswire.com/digital-experience/how-human-computer-interaction-can-help-marketers/

Okesola, O. J., Adebiyi, A. A., Owoade, A. A., Adeaga, O., Adeyemi, O., & Odun-Ayo, I. (2020, July). Software Requirement in Iterative SDLC Model. In *Computer Science On-line Conference* (pp. 26-34). Springer, Cham.

Oliveira, F. de S., & Barbosa, V. C. (2016). A note on counting independent terms in asymptotic expressions of computational complexity. Optimization Letters, 11(8), 1757–1765. doi:10.1007/s11590-016-1092-7

Olson, G. M., & Olson, J. S. (2003). HUMAN-COMPUTER INTERACTION: Psychological Aspects of the Human Use of Computing. *Annual Review of Psychology*, 491-516.

Onwuegbuzie, A. J., Dickinson, W. B., Leech, N. L., & Zoran, A. G. (2009). A Qualitative Framework for Collecting and Analyzing Data in Focus Group Research. International Journal of Qualitative Methods, 1–21. https://doi.org/10.1177/160940690900800301

Panum, T. K., Hansen, R. R., & Pedersen, J. M. (2019, June). Kraaler: A user-perspective web crawler. In 2019 Network Traffic Measurement and Analysis Conference (TMA) (pp. 153-160). IEEE.

Pavalam M, Kashmir R, Felix K and Jawahar M (2011) A Survey of Web Crawler Algorithms *International Journal of Computer Science Issues,* Vol. 8, Issue 6 retrieved from http://www.ijcsi.org/papers/IJCSI-8-6-1-309-313.pdf

Peshave, M., & Dezhgosha, K. (2005). How search engines work: And a web crawler application (Doctoral dissertation, University of Illinois Springfield).

Proskurina, T. (2018). *Narrative visualizations: using interactive data stories in strategic brand communication.* Lund: Lund University.

Pundhir, Sandhya & Rafiq, M.. (2012). Performance Evaluation of Web Crawler.

Rennekamp, R. & Nall, M (n.d.).*Using Focus Groups in Program Development and Evaluation* in Cooperative Extensive Service. University of Kentucky College of Agriculture retrieved from https://psd.ca.uky.edu/files/focus.pdf

Rodríguez, M. T., Nunes, S., & Devezas, T. (2015). Telling Stories with Data Visualization. *NHT '15: Proceedings of the 2015 Workshop on Narrative & Hypertext*, 7-11. doi:http://dx.doi.org/10.1145/2804565.2804567

Rogers, Y. (2005). New Theoretical Approaches for Human-Computer Interaction. *Annual Review of Information Science and Technology*, 87-143.

Rogowitz, B. E., & Treinish, L. A. (1998). Data visualization :the end of the rainbow. *IEEE SPECTRUM*, 52-59.

Samadi A.,(2018)The Role of Human-Computer Interaction (HCI) in Preventing the Excesses in Attracting Customers to E-Commerce Review of Public Administration R and Management (6,1) retrieved from https://www.longdom.org/open-access/the-role-of-humancomputer-interaction-hci-in-preventing-the-excesses-inattracting-customers-to-ecommerce-2315-7844-1000239.pdf

Segel, E., & Heer, J. (2010). Narrative Visualization: Telling Stories with Data. *TVCG: Transactions on Visualization and Computer Graphics,*, 1139 –1148. Retrieved from http://vis.stanford.edu/files/2010-Narrative-InfoVis.pdf

Shukla V.&, Dharmendra R (2016) Web Crawlers and Web Crawling Algorithms A Review *International Journal of Scientific Research in Science, Engineering and Technology IJSRSET* Vol. 2 Issue 2 retrieved from https://www.academia.edu/25127230/Web_Crawlers_and_Web_Crawling_Algorithms_A_Review

Silverman D (2004). *Qualitative Research: Theory, Method and Practice*, Sage Publication (2 e.d)

Singh A V, & Mishra A. (2014) A Review of Web Crawler Algorithms *International Journal of Computer Science and Information Technologies,* Vol. 5 (5) retrieved from https://www.academia.edu/9896464/A_Review_of_Web_Crawler_Algorithms

Verma, S. (2014). Analysis of Strengths and Weakness of SDLC Models. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(3).

Waddell, T. F., Zhang, B., & Sundar, S. S. (2015). Human–Computer Interaction. *The International Encyclopedia of Interpersonal Communication*.

Weber, W. (2018). Data stories. Rethinking journalistic storytelling in the context of data journalism. *Studies in Communication Sciences 18.1*, 191-206.

Wrench, J (2001) Educational Software Evaluation Form: Towards a New Evaluation of Educational Software *V.3, N.1, pp. 34-47* Retrieved from http://www.usc.edu/education/TheSource/

Zhang N, Guo X. & Chen G (June 2008), "IDT-TAM integrated model for IT adoption," in Tsinghua Science and Technology, vol. 13, no. 3, pp. 306-311,, doi: 10.1016/S1007-0214(08)70049-X.

Kausar A. Dhaka V. Singh S. (February 2013), Web Crawler: A Review. Retrieved from International Journal of Computer Applications

Mirtaheri S. (May 5, 2014) A Brief History of Web Crawlers Retrieved from https://arxiv.org/abs/1405.0749

Haas T. (July 11, 2019) Web Crawler 101: What Is a Web Crawler and How Do Crawlers Work? retrived from https://www.webfx.com/blog/internet/what-is-a-web-crawler/

Xia J; Wan W; Liu R; Chen G; Feng Q (2019) Distributed web crawling: A framework for crawling of micro-blog data retrieved from https://ieeexplore.ieee.org/document/7446438

Xiang, L. C., Yin, O. S., & Han, P. Y. (2015, October). Intelligent web crawler for file safety inspection. In 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) (pp. 309-314). IEEE.

Zhi-hang, T., & Jun, L. (2019). Clothing Information Collection Based on Theme Web Crawler. International Journal of Advanced Networking and Applications, 10(4), 3919-3924.

Wishard L. (2012) Precision Among Internet Search Engines: An Earth Sciences Case Study. Retrieved from http://webdoc.sub.gwdg.de/edoc/aw/ucsb/istl/98-spring/article5.html