AUTOMATIC DATA MINING ON FACEBOOK

Final Report

SAJAD WAZIN

July 22nd 2022

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Garnering over 2.93 billion active monthly users [1], Meta's popular social media platform, Facebook, has become one of the easiest way to reach a public audience. This report will discuss the development of the application "FacebookWebScraper" in an effort to motivate independent researches on Facebook's suggestions algorithm and its participation in siloing* users. In particular, it will contain the goals that this project explores, the timeline of the development of the project, a detailed description of the software, its features and limitations, a sample study that outlines how to use the software's features for research and the future of the software.

* *siloing: isolate (one system, process, department, etc.) from others.*

1 GOALS

FACEBOOKWEBSCAPER The main goal that this project brings forward is that of providing an easy-to-use GUI application for independent researchers to conduct studies on Facebook. FacebookWebScraper provides the "Suggestions Scraper" option, which records an account's path through the Facebook Pages suggestions algorithm by providing two different searching methods. These two search methods are the popular tree traversal algorithms: Breadth-first search (BFS) and Depth-first search (DFS). Another tool that this application provides is the "Content Scraper", which is used for gathering data on posts that appear on Facebook pages. The latter offers the option of scraping a post's content, the comments on the post, the captions for the shares on the posts and the reactions on the post. This goal was achieved by prioritizing two software quality properties: usability and expansion. It must be easy to be use and it must be easy to expand for developers.

SILOING EFFECT As part of the AI & Society research group at McGill University, one of the compelling research areas that this project explores is the siloing effect on social media. This goal is accomplished by pairing the two tools together. By first running the Suggestions Scraper, we get a pool of page suggestions that the user is met with through their experience on Facebook. Subsequently, by running the Content Scraper, it is possible to use data mining techniques on the content that the page suggestions host. This allows a researcher to explore what type of pages are being suggested, establish any common patterns in terms of content/user-response, determine how long it takes the algorithm to suggest a specific type of page; all characteristics that can be studied to determine the intensity of the siloing effect on Facebook.

SAMPLE STUDY In this project report, a small size sample study will be presented to establish the effectiveness of the tool, the advantages and disadvantages of using this software and the types of conclusions that can be made from such a study. This study will statistically explore the topic of politics on Facebook. This goal and its completion will be further elaborated in the Experiments and Data Analysis section [7].

2 PROGRESS

2.1 Division of Labour

This project was carried independently by myself, Sajad Wazin. This means that the planning of the project, the development of the software, the documentation and the sample study was done completely by myself. However, as documented in the GitHub Repo, there is a thorough methodology for a specific type of study that can be carried out with this software. The sample study uses this methodology as a "proof of concept" to show that the software is fruitful in gathering relevant information and it displays an example of how data can be mined through the output. In particular, Tianyu Han and Ruoyun (Chloe) Zhang of the AI & Society group at McGill will be using the software and part of the methodology to carry out their own independent study.

2.2 Deliverable Schedule

Due to the restrictions imposed by Facebook [6] and the concerns with time, the deliverable schedule was updated regularly. In fact, some parts of the project were completely dropped. The following table contains a list of tasks that were accomplished, which of them were originally planned and which were later added, and the date that each task was completed.

Table 1: Deliverable Schedule

Task	Originally Planned	Date
GUI	Yes	June 4th 2022
Content Scraper	Yes	July 5th 2022
Suggestions Scraper	Yes	July 10th 2022
Encoding System	No	July 12th 2022
Sample Study	No	July 16th 2022

3 DOCUMENTATION & REPOS

The project is based in a private GitHub repository that is set to follow a local branch where I develop code for the program. Being the sole developer for this project, the repo is mainly used as a means of publishing code for progress reports and project submissions. The repo also provides access to documentation pertaining to the type of information that can be scraped with the program and how it can be used for research.

The codebase also provides a lot of documentation on proprietary functionality. This includes the searching algorithms, the scraping functions, as well as most of the internal data management. This documentation is provided in the form of Java comments. The codebase also includes documentation on features to-be-added (i.e: interaction modes).

4 SOFTWARE ARCHITECTURE

4.1 Programming Language

The programming language that was used for this project is Java. One of the main reasons behind this architectural decision is because of the expansion quality that Java offers. Being an Object-Oriented Programming language, Java allows developers to easily create data structures and interfaces that inherit from other code-bases.

SELENIUM The main library used in this project was Selenium, a web automation library. A lot of the work done for FacebookWebScraper involves manipulating data from Selenium, which, at its core, is scraped as raw HTML elements. As a result, Java's OOP features are essential to writing code that inherit from Selenium, but also to modify and process the data that has been found.

SEPARATION OF CONCERNS In order to provide a software architecture that is usable, clean and can be expanded upon, the design principle of "separation of concerns" is unavoidable. In fact, this design principle was applied to Facebook-WebScraper with two main goals: (1) avoid inappropriate intimacy between functional elements and (2) avoid cyclic dependencies between functional elements. Java makes it extremely easy to assign proper access and design well-encapsulated objects to keep data safe, secure and organized. This can be seen through not only the class division of the project, but even its package division. By separating concerns adequately, FacebookWebScraper has been designed in such a way that the "gui" package is so well-encapsulated that it can actually be completely removed from the project without removing any functionality from the application.

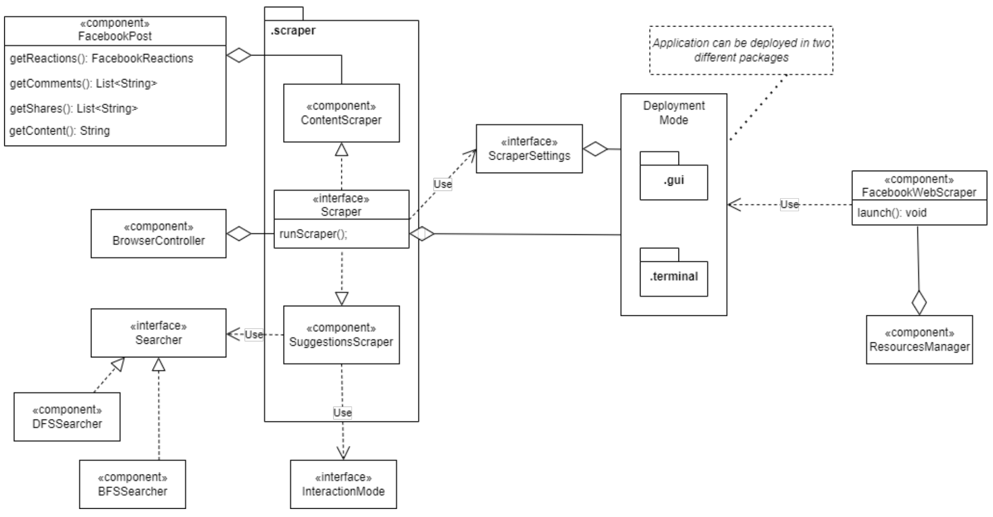

«component» FacebookPost
getReactions(): FacebookReactions
getComments(): List<String>
getShares(): List<String>
getContent(): String
.scraper
«component» ContentScraper
«interface» Scraper
runScraper();
«component» BrowserController
«interface» Searcher
«component» SuggestionsScraper
«component» DFSSearcher
«component» BFSSearcher
«interface» InteractionMode
Use
«interface» ScraperSettings
Application can be deployed in two different packages
Deployment Mode
.gui
.terminal
«component» FacebookWebScraper
launch(): void
«component» ResourcesManager

Figure 1: A UML Functional Structure Model containing the important functional components of FacebookWebScraper

4.2 Libraries

The choice of libraries in this project is the compromise between the size of the application's deployment package and the amount of well-documented and tested external code that can be used. Thus, the libraries used in the project are only the essentials and were used as a foundation to be built upon.

4.2.1 Selenium

Selenium is the main driver in the development of the application. It contains all the code that allows to start up and control a web browser. FacebookWebScraper uses it to find "WebElements" that contain the relevant data. It mainly does so by using the page's XPath syntax to locate specific CSS properties in conjunction with a mix of JavaScript and Selenium's built-in web-interaction features to navigate different WebElements. The main features used from Selenium are the following:

1. Locating WebElements by ID and XPath expressions
2. Clicking on WebElements
3. Running JavaScript commands and passing WebElements as arguments
4. Implicit waiting (sleep)
5. Handling different tabs
6. Initializing the browser with custom settings

4.2.2 JavaFX

JavaFX is a framework for creating GUI applications in Java. Using JavaFX, the default way to launch and use FacebookWebScraper is through the GUI. It provides an interface for users to program different scrapes on Facebook without any coding background. Similarly, it also allows to sanitize the scraping settings by restricting what the user can input.

4.2.3 *JSON*

JSON provides an easy-to-use and readable file format to save the data gathered by the application. The choice of JSON as the file format is mainly motivated by its tree-like structure, as well as its near universal support among programming languages.

4.3 Maven

Maven is an essential tool for the development of this application. It is a build automation tool that, in this case, is used for dependency management and artifact generation. It is used to fetch the required versions of Selenium, JavaFX and JSON for the project, without distributing the libraries as part of the code-base. Similarly, by using the Maven Compiler Build plugin, developers can run tests by simply compiling the project and running it through their IDE of choice*. In particular, this compiler plugin allows to easily set distinct Java versions for the source code of the project and the compiled target artifact.

Another important aspect of Maven is the Maven Shade plugin that is used to generate an executable ".jar" artifact for FacebookWebScraper. This Shade plugin allows to create "fat jars", which are Java artifacts that are compiled and packaged with all relevant dependencies and libraries. In this case, since Java no longer packages JavaFX for SDK versions after 11 and because Selenium and JSON are not part of the Java SDK, these dependencies must be built into the artifact to run properly. Similarly, since JavaFX requires inheritance from its "Application" class for it to launch, Maven can be used to direct the compiler to the main "Application" class.

4.4 Batch Sequential Architecture

This program follows a simple Batch Sequential Architecture. The Batch Sequential Architecture usually requires a series of unordered "pipes" and "filters". It is focused around data processing, from one input, through a series of different algorithms, with one output. FacebookWebScraper respects this architecture very closely, as the output of the Suggestions Scraper can become the input of the Content Scraper [5.4.2], and filtering the output of the Content Scraper with simple scripts can produce an input for the Suggestions Scraper. In particular, this Batch Sequential Architecture was used in the sample study [7.3], initially starting with a list of "root" pages that was fed into the Suggestions Scraper, output of which was fed into the Content Scraper, which, finally was parsed through a Python script to conduct statistical analysis. This type of procedure requires little to no human interaction, and can be repeated very easily, given only one input file.

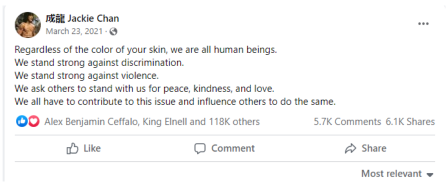
* The GitHub repository contains a project-specific ".idea" folder respective to IntelliJ, but any IDE can be used to work on the project.

5 FACEBOOKWEBSCRAPER

The software will come prepackaged in a Java executable application. It contains a plethora of features and has been designed to be easily expanded*.

5.1 Content Scraping

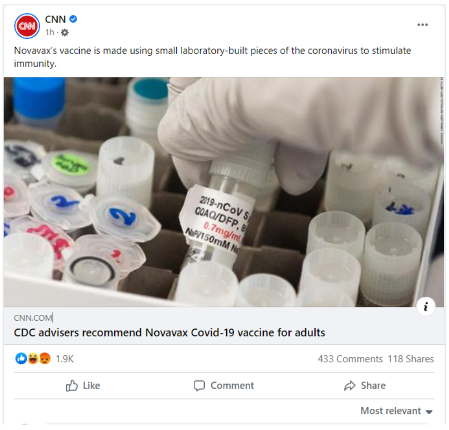
The content scraper is one of the main features offered in this application. Given a list of links for Facebook pages, users can scrape any given number of article-style posts on a Facebook page.


成龍 Jackie Chan
March 23, 2021 ·

Regardless of the color of your skin, we are all human beings.
We stand strong against discrimination.
We stand strong against violence.
We ask others to stand with us for peace, kindness, and love.
We all have to contribute to this issue and influence others to do the same.

Alex Benjamin Ceffalo, King Elnell and 118K others 5.7K Comments 6.1K Shares

(a) Post containing only text

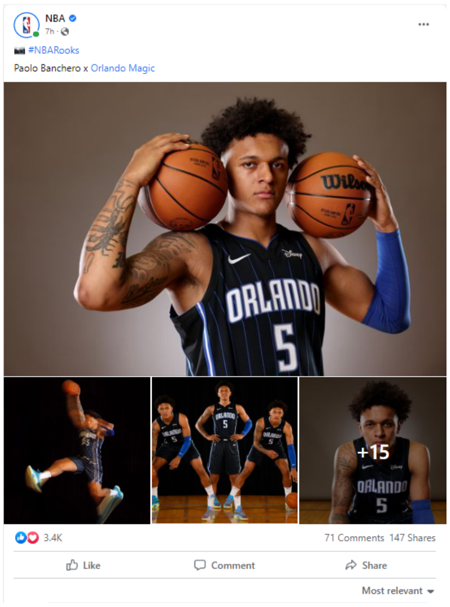

CNN
1h ·

Novavax's vaccine is made using small laboratory-built pieces of the coronavirus to stimulate immunity.

CNN.COM
CDC advisers recommend Novavax Covid-19 vaccine for adults

1.9K 433 Comments 118 Shares

(b) Post containing a link


NBA
7h ·

#NBARooks
Paolo Banchero x Orlando Magic

3.4K 71 Comments 147 Shares

(c) Post containing pictures

Figure 2: Examples of article-style posts

The content scraper is able to find the unique link that leads to a post by searching the HTML contents of a given Facebook page. Once it has found the link, the browser will open the post in a new tab to have access to all the data provided on that post, but also to not lose the feed of posts that was loaded on the Facebook page. After opening the post, the content scraper will then gather all the data that it has been programmed to scrape.

* For more info on how to use the program, visit this link.

The content scraper settings can be configured to do the following:

- Scrape the content of the post
- Scrape a determined amount of comments on a post
- Scrape the reactions to a post
- Scrape a determined amount of shares on a post
- Use multithreading to scrape multiple pages at once
- Handle any errors caused by Facebook

It is important to note that Facebook can identify when and if certain interactions are likely to be caused by bots. This concept will be further explained in section [6]. For this reason, although multithreading does function properly, it also increases the risks of being flagged as a bot. Similarly, conducting a scrape for elongated periods of time also makes it more likely to have certain permissions temporarily revoked.

5.2 Suggestions Scraping

Given Facebook's tree-like structure for providing suggestions, it is only appropriate to navigate suggestions using tree search algorithms. Two of the most popular search algorithms are the Breadth-first search (BFS) and Depth-first search (DFS) algorithms. This section will explain how the algorithms function and the necessity of implementing both in the software. To illustrate these concepts, imagine that each node in the following tree represents a Facebook page.

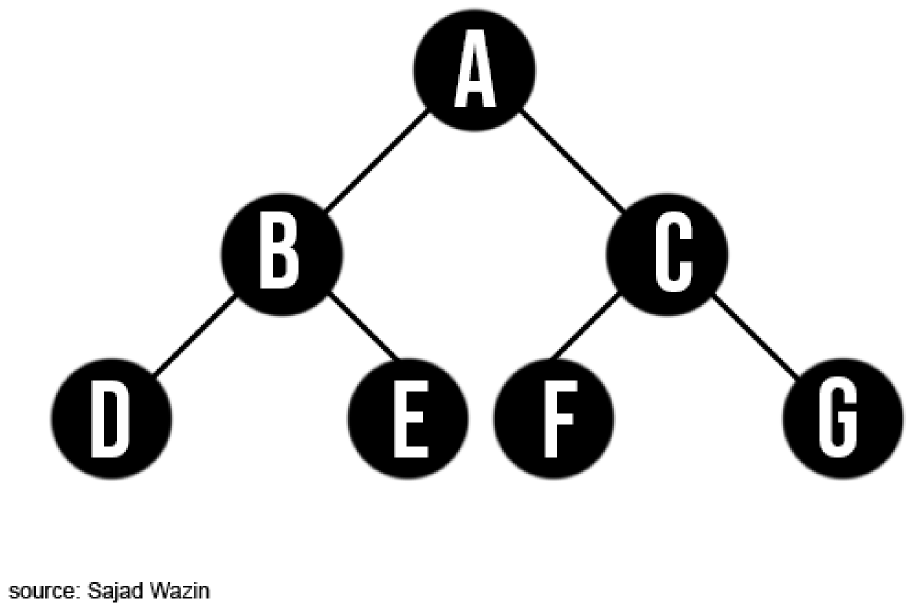


Figure 3: A tree-like representation of Facebook suggestions*

* It is important to note that not all suggestion trees on Facebook are acyclic. This was accounted for in the software to avoid infinite loops.

5.2.1 *BFS*

As the name suggests, a breadth-first search consists of searching through all items in a specific level before moving onto a subsequent level. In the case of Facebook suggestions, these levels can be thought of as "depth" or "distance from the root". Using the example provided in the above tree [Fig. 3], a breadth-first search of these suggestions would result in the following order: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$. The reason why such a search method must be employed is because it is most likely to emulate the way a user likes pages over time. In particular, as a user likes a page and witnesses its content, then it is more likely to be exposed to the related pages and their contents. Thus, this search method explores how the suggestions algorithm responds to a more natural way of liking pages on Facebook. The way this works in the software is by first liking all the immediate suggestions from the root. While doing so, the software will store all of the pages that have been recommended upon liking all of these immediate suggestions in the Queue for the next depth. The software will then repeat this process of liking pages, storing suggestions, and moving onto the next depth until it has reached its maximum depth, at which point the search terminates.

5.2.2 *DFS*

Contrary to the BFS algorithm, the depth-first search consists of exploring how the Facebook suggestions algorithm responds to a user "going down a rabbit-hole", which usually happens over a shorter period of time. This algorithm is better explained by classifying page suggestions as "children" and "siblings". A page's immediate suggestions would be a page's "children", and any two page that are suggestions of the same page would be "siblings". In reference to the given example [Fig. 3], nodes B and C would be siblings to eachother, and both are children of A. Then, the algorithm works as follows. By going in left-to-right order, the subtree of a node must be completely searched before moving to its sibling. Using the example, a DFS of the tree would result in the following order of visited nodes: $A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow F \rightarrow G$. The property that this searching method studies is the reaction of the suggestions algorithm to a user going down a rabbit-hole of a specific type of content. For example, let A, B, D and E share all very similar content, but all of which be extremely different from C. This searching algorithm allows to see if C's suggestions will be part of the more appealing content for the user (similar to A, B, D and E), or if they will only be similar to C and unaffected by the user's past actions. This is implemented by using a Stack data structure, which allows each node to recursively search once for each of its children.

5.3 GUI

FacebookWebScraper offers an easy to use GUI that doubles as an input validator. It allows users without coding experience to scrape without going through the hassle of cloning the repository and repackaging the software. The GUI has the following features:

- Login field to set the Facebook credentials
- A settings configuration dialog to setup a scrape
- File choosers to set the input and output
- A read-only console to keep track of the scraping status
- A "Cancel Task" button that will safely stop scraping as soon as the current post/page scrape is over

- A "Dump Memory" button that will dump the current memory/log files without terminating the task
- A "Kill Process" button that will dump the current memory/log files and terminate all tasks

These are mostly software quality of life features that ensure the program is used properly and safely. The "Cancel Task", "Dump Memory" and "Kill Process" features also open the door for proxies and multiple logins solutions [9] to be added in the future, in which case each multithreading becomes much more viable and these options can be applied individually to each thread to more flexibly scrape data.

5.4 Encoding

The encoding solution is a simple system that is mainly based around the "JSONObject" and "JSONArray" interfaces provided by JSON. These interfaces allow for a tree like structure of the data, which can translate to data structures conforming to the "Dictionaries" and "Lists" ADTs respectively.

5.4.1 *Content Scraper Encoding*

The results of running a content scraper are encoded as follows: the content, comments, reactions and shares of a post are encoded into a JSONObject entry identified by the post URL. Each post entry is encoded in a JSONArray identified by the "posts" entry. Each "posts" entry is encoded into a "page" JSONObject that also contains the URL to the page. Finally, each page is encoded in the main "pages" JSONArray. This gives rise to the following structure:

Pages (Array) -> Page (Object) -> Posts (Array) -> Post (Object)

One of the main reasons why this encoding solution is optimal is because data points do not need to be ordered and are not proprietary to the file they reside in. Therefore, this allows a user to merge or separate the output data very easily. This becomes very useful if somebody needs to run multiple scrapes over a long period of time (days, weeks, e.t.c) or if a user needs to parse through a very large dataset.

5.4.2 *Suggestions Scraper Encoding*

The results of a suggestions scraper are encoded in two different ways. One of them follows the JSON format, while the other is a plaintext ".txt" file.

TREE-LIKE STRUCTURE This method uses a recursive "SuggestionNode" interface. In the software, a SuggestionNode has only three data fields: (1) its URL as a String, (2) the depth at which it has been found as an Integer and (3) its children which are of type "SuggestionNode". Using this, it becomes very easy to create an encoding system for suggestions; simply let each suggestion be a JSONObject containing the URL and depth as JSONObjects, and the children as a JSONArray.

RAW OUTPUT The raw output file of a suggestions scraper is an unordered list of Facebook page URLs. The main use case of this feature is if one needs to run a content scraper on the results of a suggestions scraper. This allows the latter task to be done sequentially, without interruption in the program or human interaction.

5.5 Resources

FacebookWebScraper uses a ".properties" file to hold all the XPath expressions used to find specific HTML elements. This achieves two things: (1) it makes the code much cleaner, easier to read and (2) it allows a user to locate which CSS classes are no longer used by Facebook while making troubleshooting much easier. It also uses Maven to package its resources (GUI assets and chromedriver).

6 LIMITATIONS

Facebook, just like most other massive social media platforms, is known to be ridden by bots [2]. In fact, nearly every fiscal quarter, Facebook disables billions of accounts, with the latest report showing 1.6 billion accounts disabled in the first fiscal quarter of 2022 [3]. To achieve this, Facebook must maintain a system that can thoroughly and accurately assess a user's account to determine whether or not it may be a real account. As a result, many limitations are imposed on users who write software that makes use of Facebook*, and, in particular, on FacebookWebScraper.

6.1 Content Scraping

POST IDENTIFIERS To open a post with FacebookWebScraper, the software must be able to find the unique post identifier and then follow that identifier to have access to a webpage containing all the post data. A Facebook user can find the post identifier of any post by simply clicking on the text field that contains the time at which the post was made. This, however, would be impractical for scraping as it would clear the page feed and would prevent the user from scraping more pages. An earlier version of FacebookWebScraper was able to extract the link from the text field using JavaScript, however, a patch was released preventing users from doing this. As a result, FacebookWebScraper now looks for the timestamp on comments, rather than on the post, as the patch had not affected the link contained in the comments. This, however, brings the downside that if a post has no comments, then it may not always be scraped.

INTERACTIONS Some interactions on Facebook have a rate limiting property, meaning that such an interaction can only be made a certain number of times in a specific time period before the user is no longer allowed to make such interactions. In the specific concern of content scraping, this would include opening the "Reactions" prompt. From personal testing, if a user looks at the reactions on ~200 posts in less than an hour, then the user can no longer look at reactions for the next 24 hours. This means that if a scrape is targeting a maximum that will surpass 200 posts, it is not recommended to use multiple instances of the program as it would be limited much quicker. A similar scenario takes place upon opening too many posts in different tabs, with the latest test getting an account blocked after opening ~600 posts in a day.

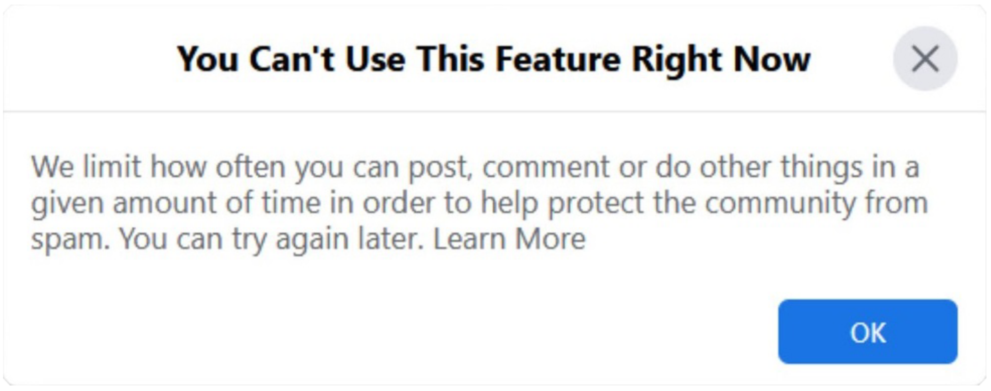
6.2 Suggestions Scraping

GROUPS This is not a security measure taken by Facebook, but rather a limitation in the efficiency of their algorithm. The original plan for the project contained a research plan for Groups as well as Pages, however Facebook's suggestions algorithm for Groups is very inefficient. In nearly all test cases, following a group would result in no suggestions. For this reason, FacebookWebScraper currently does not support group (although its "FacebookPost" interface theoretically could function on groups).

LIKING/FOLLOWING A PAGE Rate limiting is also applied to liking or following a Facebook page. As reported by Elfsight in 2017 [4], Facebook should allow a

* Such limitations may be avoided with proper approval from Facebook or with access to API permissions.

user to like approximately 5000 pages in 24 hours. However, personal testing using different wait times between each page only allowed the program to like up to 100 pages in an hour*, before being preventing from liking any page for the following 24 hours. Although Facebook does allow a user to contest such a decision, after repeated offense the user may be blocked from liking altogether.


You Can't Use This Feature Right Now

We limit how often you can post, comment or do other things in a given amount of time in order to help protect the community from spam. You can try again later. Learn More

Figure 4: An example of Facebook's bot prevention warning prompts

As a result, studies making use of Suggestions Scraping are recommended to be conducted over multiple days for data gathering. If the user has access to multiple computers, running the program on different computers and different accounts can heavily speed up the process.

6.3 Accounts

LOGINS After a certain amount of excessive logins to the same account on Facebook, you may be restricted from logging into that account. This holds for accounts that have existed for years, just as well as new accounts. From personal testing, the bar seems to be set at ~1000 logins in less than a day before it is restricted for 24 hours. Granted, this amount of logins seems unreasonable for *using* the software, however for developers working on new features on the software, 1000 logins to the same may be reached relatively quickly and thus need to be wary of this fact.

SHARED IPs While reports disagree on whether or not Facebook still use IPs to keep the platform secure and free of bots, my personal testing seemed to show it does. While running tests accounts that had gotten restricted before, a certain amount of restrictions on each account could start affecting the others. This was brought up in a test where, using only one account, after it got restricted, then all my other accounts would also get restricted. It is important to note that my personal account, which had not been used for testing in over a month at this time, did not get restricted. This seems to show that using multiple accounts, on the same IP address, around the same time period, that are deemed to be bots by Facebook can actually affect all the accounts at once.

NEW ACCOUNTS When creating a new account, the user must sometimes wait before having access to all features on Facebook. Some of these features are, but not limited to, having access to Facebook suggestions and liking pages. This would make such an account unable to be used for Suggestions Scraping. However, usually after 24-48 hours, new accounts have access to these features. It is also important to note that new accounts seem to be flagged more quickly.

* This was tested by setting the program to sleep for a minimum of 30 seconds to a maximum of 5 minutes between each page like.

6.4 Obfuscation

MINIFIED CSS Facebook uses minification to make its CSS and JavaScript files smaller and easier to download for the user. The minifier they use (likely part of their own HHVM PHP compiler) also obfuscates CSS class identifiers for nearly all HTML elements. Although locating an HTML element and its CSS class is not necessarily difficult, the issue comes up on rebuilds of Facebook. During testing, CSS class identifiers seemed to change from time to time, usually adding or removing one or many identifiers from an HTML element. This means that the XPath expression that the scraper uses may need to be updated if a Facebook rebuild changes its CSS class identifiers.

7 EXPERIMENTS AND DATA ANALYSIS

7.1 Goal

It is no secret that Facebook makes it very easy to share your opinions with people. The keyword in that sentence is "people". On most other social media platforms, people usually have an "online identity". Facebook, on the other hand, is about people and their real lives. As a result, it is very important that such a media platform must remain impartial in the propagation of people's opinions.

A great example of this would be American Politics, which is known to be a controversial subject for many. It is no secret that people usually take to Facebook to share their political opinions upon certain events. However, reports have come out alleging that Facebook silences left-leaning political opinions [5] and moreover that it favors right-leaning pages [6], [7], [8]. For this reason, this sample study is aimed at bringing out any statistical patterns that could demonstrate any type of "silencing" of left-leaning political opinions or "favoritism" towards right-leaning pages.

This will be done by accomplishing statistical analysis on page suggestions to determine if suggestions on politically charged pages tend towards a specific wing of the political spectrum. As a follow-up, there will be a statistical analysis on reactions to every suggestion to determine if Facebook silences contradicting or controversial opinions.

7.2 Methodology

This section is concerned with explaining how the results from FacebookWebScraper can be used in statistical analysis. There are at least two type of statistical analyses that can be conducted with the output of the software: (1) analysis on the siloing effect of Facebook and (2) analysis on the radicalization of opinions on Facebook.

SILOING EFFECT The siloing effect on Facebook can be studied by determining if the suggestions over a long period of time follow a certain trend. This can be done by manually assigning a "type" to each suggestion that the software encounters and then determining if any specific type is an outlier. In this particular study, suggestions were gathered by using a mix of the BFS [5.2.1] and DFS [5.2.2] searching algorithms. Once the suggestions are encoded, each page is assigned a type based on their political affiliations. This establishes a clear delimitation for pages based on the political spectrum; either a page is left-leaning, right-leaning, neutral or not politically charged.

RADICALIZATION OF OPINIONS To determine if an opinion is being radicalized on Facebook, one must be able to establish a clear pattern of "propaganda" of an opinion or "silencing" of a counter-opinion. This study focuses on the former. By using Natural Language Processing, and more particularly, sentiment analysis, a user is able to determine how much of an audience is in agreement or disagreement with an opinion. Furthermore, Facebook already offers an interface for sharing your opinion in the form of reactions. As a result, a point system can be established to assign a value to each type of interaction. In this study, the point system that was used assigned a single point to reactions and 10 points to comments. With TextBlob [9], sentiment analysis can be used to determine the polarity of a string of words. Based on this, comments can be classified in their respective sentiment group.

Table 2: Sentiment Analysis of Comments

Polarity (between -1 and 1)	Sentiment
-1 <= pol < -0.333	Negative
-0.333 <= pol <= 0.333	Neutral
0.333 <= pol <= 1	Positive

A similar concept is applied to reactions. Determining which reactions are positive, negative and neutral is much more simple given that reactions usually convey emotions, not opinions. As a result, reactions are classified the following way:

Table 3: Sentiment Analysis of Reactions

Reactions	Sentiment
Sad Emoji, Mad Emoji	Negative
Wow Emoji	Neutral
Like Emoji, Hearts Emoji Care Emoji, Laughing Emoji	Positive

With these classifications, it becomes easy to evaluate how an audience feels about most Facebook posts.

7.3 Sample Study

The sample study was conducted in three different parts. It is first comprised of a "control" study, which scrapes data on posts from a random selection of some of the most popular Facebook pages. This random selection contained pages such as media platforms (YouTube, NetFlix, National Geographic, BuzzFeed), corporate brands (Coca Cola, McDonalds), movie and television franchises (Harry Potter, The Fast Saga, The Simpsons), celebrities (Selena Gomez) and other types of pages. It serves the purpose of defining what an "average" experience should reflect on Facebook, on pages that most people know of or follow. These pages are unlikely to cause controversy and create a divide between its followers, thus the response to its content is a good reflection of the audience's immediate reaction to the content rather than a bias opinion from an external influence. The data that is scraped will then be turned into a statistic defining the "average sentiment analysis for a Facebook page". It will be used as a baseline for comparison with the following results.

Subsequently, a study on handpicked left-leaning and right-leaning pages will be lead. Similarly to the control, this study will involve scraping data on posts from these pages and turning it into a statistic that defines the "average sentiment analysis for left-leaning/right-leaning pages on Facebook".

Finally, using the Suggestions Scraper, a list of pages will be gathered by searching through Facebook page suggestions using left-leaning/right-leaning pages as the roots of the search. Then, using the content scraper, data will be gathered to determine a statistic that defines the "average sentiment analysis for suggestions from left-leaning/right-leaning roots". It is important to note that it cannot be expected that all these suggestions will be as politically charged as the roots. A consequent analysis will be conducted to manually determine how many suggestions had political affiliations and on which side of the spectrum they reside.

The data that will be gathered should allow to make conclusions on the following questions: (1) does Facebook favor right-leaning pages over left-leaning pages and (2) are left-leaning opinions silenced as opposed to right-leaning opinions.

7.3.1 *Control Study*

The control study was conducted on 25 pages and consists of 308 post scrapes. These pages are some of the most popular pages of different categories, including: sports, entertainment, celebrities, brands, e.t.c. From the results of this control study, it can be shown that the average sentiment analysis of a Facebook page has the following division:

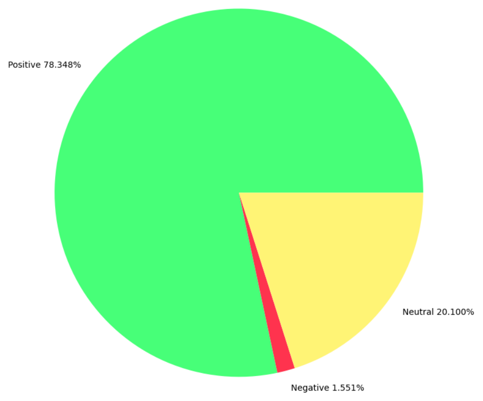
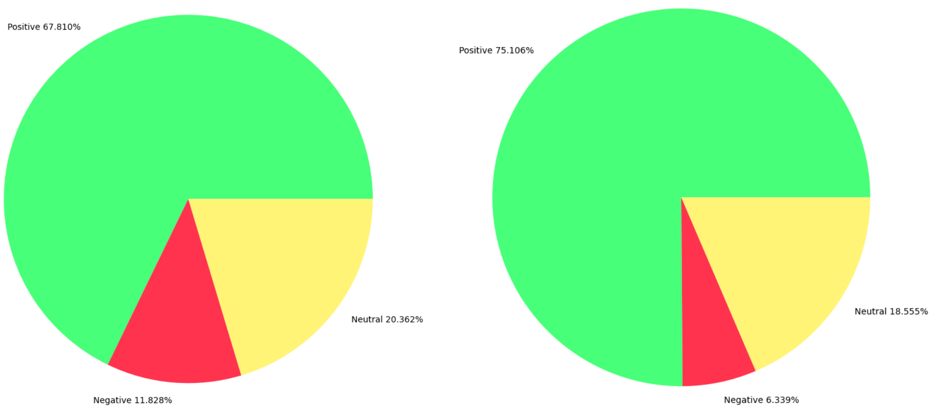


Figure 5: Average sentiment analysis on a Facebook post

7.3.2 *Handpicked Pages*

The study on handpicked pages was done by performing sentiment analysis on 10 popular left-leaning pages and right-leaning pages. These pages are mostly media outlets and internet personalities. The output consists of the analysis of 298 posts and 220 posts respectively.

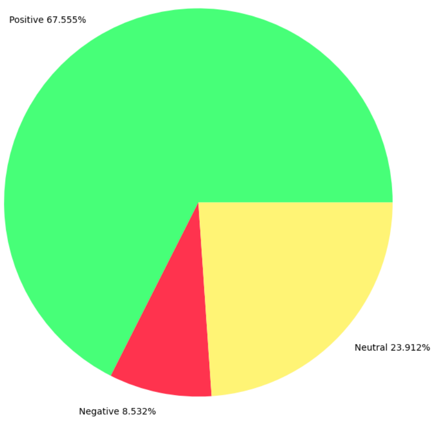


(a) Left-leaning Facebook pages **(b)** Right-leaning Facebook pages

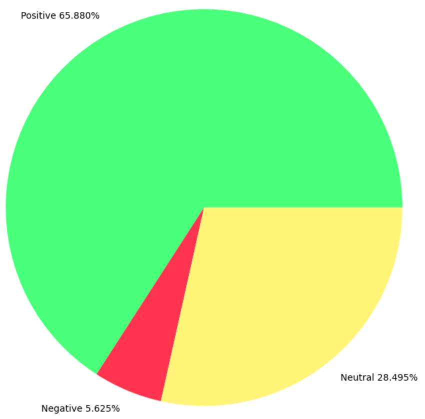
Figure 6: Average sentiment analysis on handpicked pages

7.3.3 *Page Suggestions*

The study on page suggestions was done by using 3 of the handpicked left-leaning and right-leaning pages as roots for the suggestions scraper. Then, the output of the suggestions scraper was used as input for the content scraper to perform sentiment analysis. It is important to note that in the output from the left-leaning roots, only 3 out of the 22 suggestions were found to be politically charged (all of which were left-leaning as well). On the other hand, for the output from the right-leaning roots, 12 of the 22 suggestions were found to be politically charged (3 left leaning and 9 right leaning). The sentiment analysis was performed on 261 posts from suggestions of left-leaning roots and 312 posts from suggestions of right-leaning roots.



(a) Left-leaning Facebook roots



(b) Right-leaning Facebook roots

Figure 7: Average sentiment analysis on suggested pages

7.4 Results

While the data seems to show that, in general, politically affiliated pages tend to have less positive results than an average Facebook page, there is little evidence to support the claim that left-leaning opinions are being silenced. In fact, while looking at the percentage of positive response to pages, Figures 6.(a), 7.(a) and 7.(b) have nearly identical results. If one was to assume that a positive response would be a show of agreement to the page, this would mean that people who share the same opinions as the given political party the page promotes are not being silenced in sharing their beliefs on the page. However, Figure 6.(b) could be interpreted as providing some evidence to the fact that right-leaning opinions are being advantaged on Facebook. This, paired with the added result of the suggestions scraping on right-leaning roots demonstrating a bias for suggesting politically affiliated pages, could indeed demonstrate some sort of favoritism. However, given the sample size of this study, the data gathered is not conclusive in showing any type of left-leaning opinions being silenced, or any definite proof that right-leaning pages experience favoritism.

7.5 Procedure Revision

The procedure used in this study does have room for improvement. The obvious first step towards a better result would be increasing the sample size. Taking in consideration that this study was conducted in the span of a week, in parallel with the development of the software, there was no other option but to use a small sample size for the study. By increasing the amount of pages and posts, it becomes much more suitable to found a scientific conclusion on the evidence that was gathered.

Another way the procedure can be improved is by redefining the point system established in [7.2]. While it makes sense that comments should be worth more points than reactions because of how much more dedication it takes to leave a comment, along with the fact that a user has more freedom of expression in the comment section, attributing it a value of "10" is totally arbitrary. In fact, a much better course of action would be to attribute a value to comments based on the page/post that is currently being scraped. One specific way to do this is by attributing a post-specific value to comments that, after summing up all the comments, would contribute as much "weight" as reactions.

Finally, Natural Language Processing can be very tricky. As this project was my first interaction with the subject, it was very difficult to find an appropriate way to incorporate it in the analysis process. The main flaw in the usage of sentiment analysis is that it does not take into account the content of the post. Therefore, it is not always true that a comment with a positive polarity is in agreement with the content on the post. For example, a post with the following caption: "I am extremely saddened by the Boston Celtics' loss in the NBA 2022 Finals" would have -0.125 polarity based on TextBlob. Now, imagine scraping the following comment on this post: "I am so happy that they didn't win!". The latter would have a 0.9 polarity based on the sentiment analysis. While the post's content is neutral and the comment would be positive, it does not necessarily mean that they are in agreement with eachother. However, it is assumed this type of implied disagreement is not very common in discourse. For this reason, given a greater sample size, this may be insignificant.

8 CONCLUSION

This report serves the purpose of demonstrating what has been done in the efforts of producing an easy-to-use application that can be deployed in research. In general, I believe the development of the application has been a success. FacebookWebScraper has two main purposes: discover patterns in the suggestions algorithm and conduct studies on Facebook pages. While the study in section [7] was scientifically inconclusive; the outcome may have been different, had the results come from 10,000 ~ 20,000 posts and 1,000 pages as opposed to 1,399 posts and 89 pages. It did, however, allow for a qualitative conclusion to be made: FacebookWebScraper can uncover patterns hidden in the depths of Facebook's algorithms.

This is not to say that the project cannot be improved. In fact, there are many things that could have been done differently. First of which could've been using a library such as Scrapy [10] instead of Selenium. The main reason why Selenium was used is because, as of July 2022, all projects taking place in the AI & Society research group uses Selenium. For the sake of consistency among current and future project contributors, it seemed appropriate to stick to the library. However, Selenium is generally better for automated web-testing, while Scrapy is optimized for scraping data.

Another point of improvement would've been to get special access to the Facebook Graphs API. In fact, this project is highly motivated by my failure at doing so. Although I managed to be attributed access to the API, using it for research purposes is against the Terms of Service. Upon discovering that, my solution was to develop FacebookWebScraper, which emulates some of the features of the Graphs API by scraping publicly available data. Another solution would be to try to get in contact with Facebook Support for Developers to get special authorization to use the Graphs API for research.

Nevertheless, if this project is to be picked up by future contributors at the AI & Society group, it would be interesting to put a larger focus on the research aspect of the project. With this being the current group's first and only attempt at tackling Facebook, much of the time allocation was spent single handedly developing the application, leaving very little time to gather enough data for significant research. Now that the framework is in place, the next course of actions should be to use the application to get access to data for research on different topics and grow a better understanding of Facebook's artificial intelligence algorithms and how it affects its users.

9 FUTURE OF THE SOFTWARE

For what is of the software, there are two main features to add that could greatly improve its performance.

VIDEOS & WATCH As mentioned in section [5.1], the application may only conduct content scraping on article-style posts. In testing, this limitation could usually be a "hit-or-miss". Some pages (i.e: Harry Potter Movie page) would only contain Facebook videos, which have a different layout from the article-style posts and thus could not be scraped unless a different scraping interface is written. Similarly, the Facebook Watch platform offers yet a different layout from the videos and article-style posts. For this project, these were deemed out of scope as scraping the contents of a video would have been unfeasible with the level of Natural Language Process I have. However, there exists libraries that will transcribe any given video, thus this could be a feature worth adding.

PROXIES & MULTIPLE LOGINS In terms of performance, the largest bottleneck is Facebook itself. By limiting the user's interactions [6.2], Facebook only allowed me to scrape at most 200 post in less than an hour using multithreading, which resulted in my account being temporarily disabled for 24 hours. While running only one instance of the application allowed me to scrape over 600 posts in less than 24 hours before being temporarily disabled, the total time taken for this scrape was over 7 hours. One way to avoid that is by changing the login solution. Currently, the software uses a "credentials" method, which prompts the user to put in a single email and password pair that will be used to run the scraper. No matter how many instances the user decides to run, each instance will log into the same account. By changing the login solution to take an input file with multiple "username:password" pairs, the program should automatically run one instance of the scraper per valid login. This would not change anything in terms of rate limiting per account, each account is still subject to the same limits. However, if all of these accounts are running concurrently, then the total amount of posts scraped can be greatly increased. In a similar fashion, as suspected in section [6.3], it is possible that Facebook uses IPs to determine if multiple accounts are being used by the same person. To avert that from happening, a proxy or VPN solution could be implemented such that each instance is being ran from a different virtual location. In fact, Selenium can be ran from Docker, which could make the implementation of such a feature easy for those experienced with the platform.

REFERENCES

[1] Meta Platforms, Inc. (2022) Meta Earnings Presentation Q1 2022. [Online]. Available: https://s21.q4cdn.com/399680738/files/doc_financials/2022/q1/Q1-2022_Earnings-Presentation_Final.pdf

[2] R. Lever. (2019) Fake facebook accounts: the never-ending battle against bots. [Online]. Available: https://phys.org/news/2019-05-fake-facebook-accounts-never-ending-bots.html

[3] S. Dixon. (2022) Global number of fake accounts taken action on by facebook from 4th quarter 2017 to 1st quarter 2022. [Online]. Available: https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter/

[4] Elfsight. (2017) Facebook Limits (groups, friends, invites etc) and Account Blocks. [Online]. Available: https://elfsight.com/blog/2017/05/facebook-limits-and-blocks-avoiding-account-bans/

[5] A. Damon. (2021) Facebook purges left-wing pages and individuals. [Online]. Available: https://www.wsws.org/en/articles/2021/01/23/pers-j23.html

[6] A. Thompson. (2020) Why the right wing has a massive advantage on facebook. [Online]. Available: https://www.politico.com/news/2020/09/26/facebook-conservatives-2020-421146

[7] B. Allyn. (2020) Facebook keeps data secret, letting conservative bias claims persist. [Online]. Available: https://www.npr.org/2020/10/05/918520692/facebook-keeps-data-secret-letting-conservative-bias-claims-persist

[8] S. Ellison and E. Izadi. (2021) Facebook allowed conservative outlets to spread misinformation. [Online]. Available: https://www.washingtonpost.com/business/2021/10/26/conservative-media-misinformation-facebook/

[9] S. Loria. (2020) Textblob: Simplified text processing. [Online]. Available: https://textblob.readthedocs.io/en/dev/

[10] Zyte. (2022) Scrapy | a fast and powerful scraping and web crawling framework. [Online]. Available: https://scrapy.org/