|  |  |
| --- | --- |
| School of Electronic Engineering and Computer Science | **Interim Report**  **Programme of study:**  MSci FT Computer Science  **Project Title:**  **Hate Speech Filtering Web Extension**  **Supervisor:**  Arkaitz Zubiaga  **Student Name:**  Yulong Cui  Date: 29/11/2021 |
| Final Year  Undergraduate Project 2021/22 |
| QMLogo |

Abstract

This project aims to solve the problem of online textual racial abusive contents using a web extension that identifies and filters out hate speech on webpages using Natural Language Processing techniques, thus minimize the user’s exposure to textual racial abusive content on the internet.

**C**

**ontents**

[Chapter 1: Introduction 4](#_Toc89041467)

[1.1 Background and Motivation 4](#_Toc89041468)

[1.2 Problem Statement 4](#_Toc89041469)

[1.3 Objectives 5](#_Toc89041470)

[1.4 Report Structure 5](#_Toc89041471)

[Chapter 2: Literature Review 6](#_Toc89041472)

[2.1 Python & related libraries 6](#_Toc89041473)

[2.2 Machine Learning 7](#_Toc89041474)

[2.3 Existing Systems 8](#_Toc89041475)

[2.4 Functional requirements 9](#_Toc89041476)

[2.5 Non-functional requirements 9](#_Toc89041477)

[2.6 Risk assessment 10](#_Toc89041478)

[2.7 Original Project Plan 11](#_Toc89041479)

[2.8 Amended Project Plan 11](#_Toc89041480)

[Chapter 3: Methodology 12](#_Toc89041481)

[3.1 GUI design 12](#_Toc89041482)

[3.2 Sequence diagram 14](#_Toc89041483)

[Chapter 4: Presentation 16](#_Toc89041484)

[4.1 Source Code 16](#_Toc89041485)

[Chapter 5: Evaluation 16](#_Toc89041486)

[Chapter 6: Conclusion 16](#_Toc89041487)

[Additional Appendices (as needed) 16](#_Toc89041488)

[References 16](#_Toc89041489)

# Introduction

## Background and Motivation

Racism is one of many expressions of our evolved capacity to live and work in groups. The very human tendency to identify with an ‘us’ defines the broader ‘them’.[[1]](#endnote-1) The most common way of racial abuse is textual based, whether it is being said in real life or a being sent as replies on social media.

With the world moving gradually into this virtual environment online where billions of people around the world can connect with each other at their fingertips. Unfortunately, racially motivated abuses are very common nowadays on the internet. Mediums ranging from comment sections on big social media platforms to ‘mis-worded’ news articles are all guilty of spreading hate speech contents.

The aftermath of the Euro 2020 Final was a prime example of big social media platforms not being proactive enough to prevent online racial abuse. Twitter and Instagram said they had removed thousands of racist comments in the wake of the Euro 2020 final, more than 2000 racists tweets/posts were made towards the three England players since the defeat on the eve of July 22nd. 207 accounts were investigated by the NPCC (National Police Chiefs Council). However, the NPCC claimed that it was still waiting for these social media companies to share information about 50 of those 207 accounts on August 5th.[[2]](#endnote-2) Which clearly indicates the lack of responsibility those big social media companies are willing to take.

Research also found that experiencing racism is associated with poor mental health and, to a lesser extent, poor physical health. Racism is associated with higher rates of stress, increasing a person of colour’s risk of developing high blood pressure. In fact, the Centres for Disease Control and Prevention (CDC) report that Black people are more likely to have hypertension than any other racial or ethnic group.[[3]](#endnote-3)

## Problem Statement

Today’s online legislations and policies have not fully covered all corners of the internet world. Even if social media companies start banning accounts with great effects, new accounts can be created anonymously in a matter of seconds. Which means many people can hide behind their online identities and launch racial abuses against others without any real-life legal consequences.

Although the ultimate goal is to clear out the root of racism, it would take generations to achieve this. The internet is only growing faster and faster as time progresses. Therefore, we cannot keep relying on those big social media companies to act, we need to be proactive find a solution ourselves.

­

## Objectives

This project aims to achieve the following objectives:

* Fetch pure texts from websites and store locally (on user device) as text files using BeautifulSoup4 in Python
* Create a filtering algorithm that performs analysis on fetched text files using Python NLTK and Python Profanity library
* Create a web extension using JavaScript, Django, and HTML
* Link all pieces of code to web extension

## Report Structure

* Chapter 1: Basic introduction and motives of this project
* Chapter 2: Background research, existing systems, and risk assessment.
* Chapter 3: Use case diagram, sequence diagram with explanations. A sketch diagram of the web extension user interface.
* Chapter 4: Snippets of code to demonstrate implementation.
* Chapter 5: Analysis of end product, conclusion on whether objectives listed in 1.3 are achieved.
* Chapter 6: Consideration on what further work can be achieved.

# Literature Review

## Python & related libraries

### Python programming language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding.[[4]](#endnote-4) Combined with Python’s extensive selection of machine learning-specific libraries and frameworks simplify the development process and cut development time.[[5]](#endnote-5)

However, there are some disadvantages of using Python. Languages such as Java, C++ can perform multithreaded operations which speeds up progress when doing any kind of machine learning algorithms. Whereas Python cannot, but it can emulate it using a threading library. Python is also slower compared to other main programming languages as it requires extra steps of translations when executing programs.

Despite its flaws, as my web extension requires creating a machine learning algorithm that performs analysis on text files, Python becomes the first choice of programming language for its simplicity, dynamic typing, binding and the wide range of machine learning libraries.

### Natural Language Processing (NLTK libraries)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.[[6]](#endnote-6) As my machine learning algorithm will perform analysis on normal, day-to-day spoken words and sentences. With NLTK’s wide range of text processing functions, it allows me to classify words and sentences easily.

### Python Profanity Filter

Python Profanity Filter is a universal library for detecting and filtering profanity words and sentences. Main features[[7]](#endnote-7) include:

* Full text or individual words censoring.
* Multilingual support, including profanity filtering in texts written in mixed languages.
* Deep analysis. The library detects not only the exact profane word matches but also derivative and distorted profane words using the Levenshtein automata, ignoring dictionary words, containing profane words as a part.
* Spacy component for using the library as a part of the pipeline.
* Provide explanation of decisions.
* Partial word censoring.
* Extensibility support. New languages can be added by supplying dictionaries.
* RESTful web service

I will utilise the profanity filter library combined with the NLTK library to develop my machine learning algorithm.

### BeautifulSoup4

Beautiful Soup Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.[[8]](#endnote-8) I will be using Beautiful Soup to extract text from websites and storing it in a local text file on the user’s device in order for my algorithm to perform the necessary analysis.

## Machine Learning

### Machine Learning vs Conventional Programming

Machine learning language algorithms build a mathematical model depending on sample data. This data is known as “training data.” Using these data and algorithms, prepares predictions or decisions. Conventional programming is a manual process, which means the programmer creates the logic of the program. They need to code the rules and write lines of code manually. They provide the input data and base on the program’s programming logic; it produces the desired output. The conventional programming approach is algorithm dependent, and for a program, multiple algorithms can work. It is up to the programmer how he will design and develop the logic of the program.

Diagram

Description automatically generated

Figure 1 Differences in input and output between conventional programming vs machine learning

Machine learning comes under artificial intelligence. Artificial intelligence is an umbrella term that contains many realms like machine learning, image processing, neural networks, cognitive science, and many more. Unlike conventional programming, in machine learning language, the computer uses a pre-written algorithm and learns how to solve the problem itself. It is a more sophisticated way of solving a problem. Machine learning language is beyond algorithmic solutions; instead, it trains a machine to solve different complex tasks by itself. [[9]](#endnote-9)

Diagram

Description automatically generated

Figure 2 Layers of specific fields regarding Artificial Intelligence

## Existing Systems

### Meta (Facebook) social media platforms’ content filtering system

Meta (Facebook) and its ‘sub-companies’ all have their own content filtering system that scans through posts and comments. It filters out discriminative/hateful contents and provide user with warning notifications blocking the possible hateful contents, which also allows user to either continue to view the original content or edit their own posts/comments to comply with Meta’s Hate Speech policies.

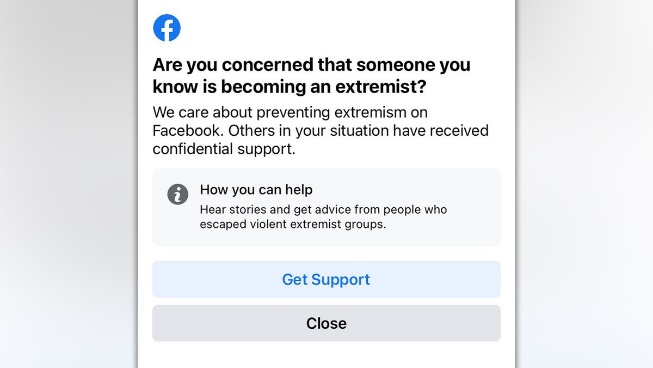


Figure 1 Example of Facebook's content filtering system

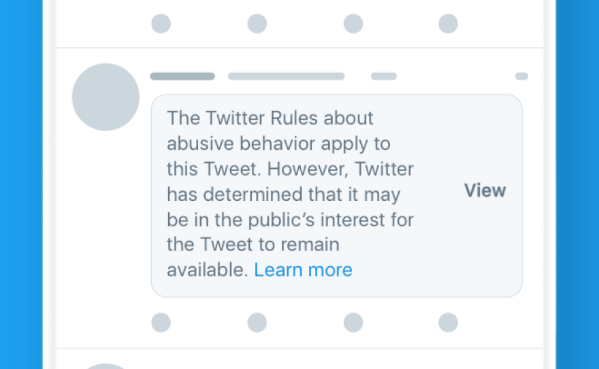


Figure 2 Example of Twitter’s content filtering systems

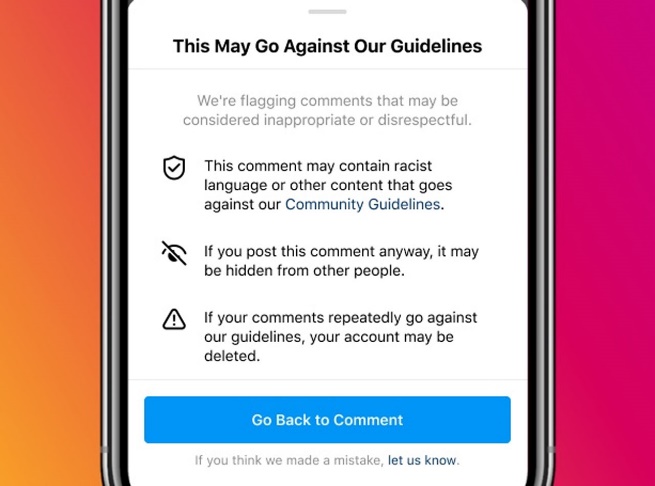


Figure 3 Example of Instagram’s content filtering system

## Functional requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Must/Want** | **Comment** |
| **FR001** | Web extension shall be compatible with most common web browsers | Must | Web extension will be made compatible with Google Chrome which is also compatible with other browsers such as Edge, Safari and Firefox |
| **FR002** | User shall be able to enable/disable the web extension by choice | Must | Web extension will have a button that switch on/off |
| **FR003** | Web extension shall block/censor potential harmful content | Must | Web extension will have a cover page blocking the entire view of web site if any potential harmful content is detected |
| **FR004** | User shall be able to choose to continue to view potential harmful content | Must | Web extension will have a button that unblocks potential harmful content |
| **FR005** | Web extension shall not store any personal details | Must | Web extension will have a disclaimer and will not store any personal details about the user, all temporary local files are deleted once user closes the web browser |

## Non-functional requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Must/Want** | **Comment** |
| **NFR001** | Web extension shall consider accessibility (colour blind mode, zoom, text size) | Want | Web extension can be made with multiple colour schemes and choices of different font sizes |
| **NFR002** | Web extension shall notify user whether its working or not | Want | Web extension can have pop-up message indicating if it is running fine or not every time user launches the web browser |
| **NFR003** | Web extension shall display a total count of how many web sites have been identified as ‘potential harmful content’ | Want | Web extension can have a stats sub page |

## Risk assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Description** | **Likelihood rating** | **Impact rating** | **Impact** | **Preventative actions** |
| **Poor time management** | Medium | High | Affect entire project timeframe, resulting in unfinished final product or a product with less standards | Utilise Gantt chart to track progress regularly, ensure sufficient progress is made on time or ahead of the time |
| **Personal illness and unforeseen events** | Low | Medium | Cause delays in final product | Ensure healthy personal schedule and diet, protect mental health |
| **Lack of technical expertise** | Medium | High | Cause delays in final product and lower standards in final product | Ensure regular contact with advisor, seek help from experienced peers if necessary |
| **Loss of code/progress** | Low | High | Loss of large portions of work if forgot to save progress/computer problems | Use cloud services such as GitHub and OneDrive for code backups |
| **Project development becomes too large to finish before deadline** | Low | Medium | Project overextends to unnecessary territories causing loss focus on main functionalities | Stick with original plan for project, minor changes can be done by myself, consult experienced peers and advisor for any major changes to project |

## Original Project Plan

## Table Description automatically generated with medium confidenceAmended Project Plan

* Extended self-deadline for ‘Extracting Textual contents’ and ‘Web extension for extracting text’ until the end and mid of December as I have been struggling with multiple deadlines and technical difficulties at home.
* Brought forward the starting point for ‘Draft Report’ and ‘Project Report’ to next week as multiple deadlines will have been passed and I want to start on the main report early.
* Everything else remain the same, main focus on coding/researching machine learning algorithms during Christmas. Integration with web extension over the next semester.

# Methodology

## GUI design

### Web extension GUI

Graphical user interface

Description automatically generated with medium confidence

The web extension is designed to be similar to an adblocker for easy-to-use purposes. It has a button to switch the filtering feature on or off and a stats section below to show a simple summary of how many websites the user visited contain potential harmful contents.

### Splash screen

Graphical user interface, text, application

Description automatically generated

The splash screen when hateful content is detected. User is warned by the splash screen of possible harmful content and has the choices of either continue to view the web page or leave the page.

## Sequence diagram

### Explanation/Methodology

The sequence diagram above shows how a user uses the Hate Speech Filtering Web Extension.

First, the user sends a filtering request to the GUI by toggling on the filtering button on the web extension. Secondly, the GUI fetches the website resources and sends it to the BeautifulSoup object textExtractor. Then the textExtractor extracts the URL of the website and sends it to a blackListed object where it checks if the website has already been blacklisted as potential harmful content.

If the website is already blacklisted, then the GUI displays a warning message with options of either view the website or leave the website and does what the user chooses to do.

If the website is not in the blacklisted websites, then the textExtractor gets all the texts from the website, stores them in a temporary local text file. The machine learning algorithm then performs analysis on the text file, determines if there are harmful content in the text file. It saves the output as training data to further improve the machine learning algorithm. If the website contains potential harmful contents, then the GUI displays a warning message with options of either view the website or leave the website and does what the user chooses to do. If it does not contain any harmful content then the GUI does not interfere with user’s normal viewing of the website.

# Presentation

## Source Code

Use the *Code* style for presenting code snippets within this report.

static public void main(String[] args) {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

}

catch(Exception e) {

e.printStackTrace();

}

new WelcomeApp();

}

# Evaluation

# Conclusion

Write your final conclusion of your project as well as any information related to future work here.

Additional Appendices (as needed)

References

1. Brooks, R., 2021. *The Origins of Racism*. [online] The Conversation. Available at: <https://theconversation.com/the-origins-of-racism-8321> [Accessed 26 November 2021]. [↑](#endnote-ref-1)
2. Perrigo, B., 2021. *Twitter Offers Transparency on Racist Abuse, But Few Solutions*. [online] Time. Available at: <https://time.com/6089289/twitter-racist-abuse-anonymity/> [Accessed 26 November 2021]. [↑](#endnote-ref-2)
3. Lewsley, J., (2020). The effects of racism on health and mental health [online]. *Medical and health information*. Available at: <https://www.medicalnewstoday.com/articles/effects-of-racism#adults> [Accessed 26 November 2021]. [↑](#endnote-ref-3)
4. What is python? Executive summary [online]. *Python.org*. [Viewed 27 November 2021]. Available from: <https://www.python.org/doc/essays/blurb/> [↑](#endnote-ref-4)
5. Python AI: why is python so good for machine learning? [online]. *Custom Software Development | Netguru*. [Viewed 27 November 2021]. Available from: <https://www.netguru.com/blog/python-machine-learning> [↑](#endnote-ref-5)
6. NLTK : natural language toolkit [online]. (no date). *NLTK : Natural Language Toolkit*. [Viewed 27 November 2021]. Available from: <https://www.nltk.org/> [↑](#endnote-ref-6)
7. Profanity-filter [online]. (no date). *PyPI*. [Viewed 27 November 2021]. Available from: <https://pypi.org/project/profanity-filter/#deep-analysis> [↑](#endnote-ref-7)
8. Beautifulsoup4 [online]. *PyPI*. [Viewed 27 November 2021]. Available from: <https://pypi.org/project/beautifulsoup4/> [↑](#endnote-ref-8)
9. How different are conventional programming and machine learning? - [online]. *Tech Blogger*. [Viewed 28 November 2021]. Available from: <https://contenteratechspace.com/how-different-are-conventional-programming-and-machine-learning/> [↑](#endnote-ref-9)