<https://qmul-ac-uk.zoom.us/j/83370948437?pwd=MHFhUEt1L1RNdGNWTVVLOUZ0akZmdz09>

Project presentation scripts

# Title:

My project is a web extension that filters out textual based harmful content on the internet.

# Background:

‘Racism’ is one of the many expressions of our evolved capacity to live and work in groups. Ever since humans developed the tendency to identify with an ‘us’ it defines the other side which is the broader ‘them’.

However, it has developed into more complicated forms of prejudices that are based on race, skin colour, gender, and sexual orientation.

The most common way of racism is textual based, whether it is being said in real life or being sent as a reply on social media.

The most recent example of this is the aftermath of the Euro 2020 final. When England lost on penalties and three England players were racially abused on social media platforms such as Twitter and Instagram. There were more than 2000 harmful tweets and posts taken down. More than 200 social media accounts were being investigated or deleted.

Research also find that racism can lead to poor mental health as it increases the level of stress, therefore, increases the risk of developing high blood pressure.

# Project Aims:

This project aims to minimize the user’s exposure to online racism, instead of relying on all the big social media companies to be responsible and filter out those harmful contents, user become more proactive and filter it out themselves.

# Project objectives:

There are four main parts to this project.

First of all is to extract textual content from websites, storing it locally as a temporary text file.

Then, create a Machine learning algorithm that performs analysis on the local text file and decide whether hateful content is present.

Finally, create a web extension that integrates the text extraction function and the machine learning algorithm.

# Computing problem:

This project is a computing problem, as the core is based on the understanding the English language using algorithms.

The English language is very complicated in terms of the structuring of a sentence, and how the meaning of one sentence can be completely different by just changing a singular word.

A word in English has a lot of variations that change it from a noun to a verb.

Also, a few words combined together becomes a phrase, however, if you change the order and combinations of phrases slightly, you can end up with completely different meanings.

Finally, the most important part is to be able to understand sentences in context.

Natural Language processing can solve all of the problems mentioned above using its text processing libraries. Combined with a Machine Learning algorithm that recognises patterns using training data and expected results, it will help to keep improving the algorithm as it is being used over time.

Moving on to literature review findings

# Literature review findings:

I have split this section into three sections, Python and its related libraries, Machine learning and different types of existing systems.

I chose python as the programming language for this project as it is an interpreted, object-oriented, high-level programming language with dynamic. Most importantly, because of its extensive range of machine learning-specific libraries and frameworks, which would simplify the development processes and shorten the development time.

Then the two related libraries: the NLTK library and the python profanity filter. Like I mentioned previously, I will be using the NLTK library to perform analysis on the text using its text processing libraries. Such as using tokenisation to understand the grammatical structure of a specific sentence, using a tagger to categorise a specific word whether it is a verb or a noun for example. I will also be using the python profanity filter to partly censor words on the webpage if the user chooses to view the webpage. The Beautiful Soup is used as a web scraper to extract texts from the web and storing locally.

Machine learning is very different from the conventional programming. As it takes results and training data as inputs, and outputs mathematical models and programs whereas conventional programming requires user written logics and unknown data as inputs, and outputs results. In this project, the machine learning algorithm comes in during the classification phrase of the Natural Language Processing. Where I would input predetermined training data such as a text file with racial abusive words and sentences and an expected result in ‘this text file should be classified as harmful’. The Machine Learning algorithm would then generate a classifier model of its first generation. Quite a few generations of the model will be needed for it to be sufficient enough to tackle real life situations such as those we’ve seen in the aftermath of Euro 2020.

# Existing systems:

There are two kinds of related systems that I have found online. The first kind is the filtering system offered by big social media companies. As you can see in the example, Meta, well previously and more commonly known as Facebook, filtering system is present on their social media platform such as Instagram and twitter. However, like I mentioned before, we must be proactive ourselves and not rely on big social media companies to filter out hate speech for us as there are billions of tweets and posts every day, it is impossible for them to filter out everything. Also, it is only available on these specific websites whereas the web extension I am working on will be available for every website.

The second kind is very similar to what this project is. These are the only two hate speech filtering web extension I found on the Google Chrome web extension store. However, as you can see, they are relatively outdated (the left one was last updated in April 2019, the right one was last updated in November 2017). I have downloaded both and tested them. Neither of them actually works, not on social media websites nor simple text files or pdf files.

Moving on to the GUI designs

# GUI Design:

My web extension’s user interface would be very similar to how an adblocker would look. There is a button to switch the filtering function on and off for the website you are currently on and at the bottom there will be a summary chart where you can see how many websites you have been to contain harmful contents.

The warning splash screen will pop up if any harmful contents are detected. It will cover the entire webpage and the user has two choices of either continue to view the webpage or leave the webpage. If the user chooses to view the webpage, then the specific bad words will be censored using the python profanity filter and the webpage will be displayed as usual. If the user chooses to leave the webpage, then the extension will either close the webpage or navigate back to the previous page.

I am using this user-friendly design to make it simpler to use and more accessible to everyone. So people that are less skilful in technology such as children, teenagers and elders can still use this web extension to protect themselves.

Moving onto the evaluations

# Evaluations:

In order to test whether my web extension works, I have split it into three main parts.

First of all is to test the text scraper on different websites, mainly for the tweets and posts on social media platforms. But also, for published documents and articles. This is to check if the text scraper is able to extract texts from different sources as my web extension is aimed to function on all kinds of different websites.

Then it is to test the machine learning algorithm and the mathematical model it has built. As I mentioned before, it would require a quite a few generations of the model for it to work properly.

Finally, after integrating the filtering function with the web extension, it is to test the full package. I would test it myself first, then upload to the Google Chrome web extension store, and ask all my colleagues to download it, test it and provide me with feedback so that I can fix bugs and improve the extension.

Moving on to project planning

# Project planning:

So far, I have completed the interim report, researched on machine learning and how to integrate it with a web extension. Over the Christmas period, I will continue to research and start implementing the machine learning algorithm.

Moving on to the risks assessment

# Risk assessment:

In terms of risks, most of them can be avoided by being proactive, for example, don’t leave tasks to the last second and aways plan ahead using the Gannt chart. But most importantly is to stay healthy during the current climate and just be sensible during the Christmas period with family gatherings.