Development of a Tool to Reduce Exposure to Hateful Content Online

Jack Bowker

School of Design and Informatics

Abertay University

BSc (Hons) Ethical Hacking, 2021

Acknowledgements

Firstly, I would like to thank my supervisor Jacques Ophoff for his expertise, support, and encouragement throughout the process of this project.

I would also like to thank my dear friends in the cozy zone Discord server, for all the nights spent working, playing, and overall trying to survive this past year with our sanity intact.

Thank you to all who completed and shared my online survey widely, I wouldn’t have been able to carry out this research without your engagement.

Finally, I’d like to thank my friends, family and partner Robbie who combined have kept me marching through this difficult year and have kept me in high spirits.

Abstract

Context

It has been observed that regular exposure to hateful content online can reduce levels of empathy in individuals, as well as affecting the mental health of targeted groups with 21% of 15-18 years olds surveyed falling victim to hateful speech online. It’s possible that Machine Learning and browser extensions could be used to identify this content.

Aim

To investigate how a browser extension can be used to assist users in reducing their exposure to hate speech online by researching existing solutions, and surveying members of the public on a proof-of-concept solution.

Method

A proof of concept browser extension was developed for the Google Chrome browser using an API and server, to explore how effective browser extensions could be in fighting hate speech online. A proof of concept extension was demonstrated to the public via a video demonstration, where participants could leave feedback regarding the usability and functionality of the extension, in order to gauge the feasibility of this approach.

Results

A proof of concept extension was developed successfully to block words using either a local word blocker or a cloud-based model. Users responded positively on the usability of the extension, as well as giving feedback regarding where the proof of concept could be improved.

Conclusion

The project demonstrated the potential for a browser extension aimed at average users to reduce individuals’ exposure to hate speech, using both word blocking and cloud-based ML techniques. However, there is potential to extend on this proof of concept project with a local ML model and extending functionality on to more websites.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

“Hate speech” is a form of targeted abuse, aimed towards an individual or group with the intent to offend or threaten (Facebook, 2021a, The Crown Prosecution Service, 2017). In online spaces, hate speech has long been an issue due to the anonymity and perceived lack of consequences of online speech, along with the ability for hateful groups to easily congregate.

It has been observed that exposure to hateful speech and sentiment towards marginalised groups can reduce levels of empathy in the wider population, with a Finnish survey reporting 67% of respondents aged between 15-18 years old had been exposed to hateful speech online, and 21% having fallen victim to such material (Oksanen *et al.*, 2014). The effect of hate speech on members of marginalised groups online is self-evident, as “From the perspective of members of the targeted groups, hate speech ‘sets out to make the establishment and upholding of their dignity’ much harder” (Barendt, 2019, pg. 543).

Several different techniques have been attempted to reduce the amount of hateful speech online, with Google funding the “Jigsaw” research wing, focused on fighting online toxicity using Machine Learning (ML). This process involves training a computer algorithm to learn and improve over time. The research involved gathering large datasets of comments and categorising toxic sentiment, offering the ability to automatically block hateful comments from sites such as The New York Times for their comment section (Google, 2019).

Even though most major social media companies have the data, resources and experience to reduce the hate on their platforms using advanced techniques such as ML, the issue is politically sensitive, with the companies cautious to ban accounts or hide posts after accusations of political censorship (Gogarty, 2020). In March 2019, CEO Mark Zuckerberg detailed Facebook’s shift towards a more “Privacy-focused” platform (*BBC News*, 2019), motivated by the public response to the Cambridge Analytica privacy scandal and spread of offensive content. This move also reduces the scope of their responsibility to moderate content on the platform, with end-to-end encryption and private groups making moderation more difficult.

These reasons are part of the reason why work is being done to give individuals control over what they see online, with Google's Jigsaw developing the experimental "Tune" browser extension, allowing users to determine the intensity of speech they're exposed to.

Extensions can be an effective place to do this detection as they're relatively accessible to users, usually installed as an add-on through an online store to extend the functionality of the browser, as well as being positioned in the web browser where most social interaction on PCs are carried out. Extensions have been used to give users control over their exposure to hate speech, with open source solutions such as Negator (Jain and Kamthania, 2020) using a locally trained Natural Language Processing (NLP) model, a type of ML that can be used to detect hateful sentiment by taking into account the context of the comment and comparing it to the comments it has seen and been trained on in the past.

1.2 Aim

The aim of this project is to investigate how a browser extension can be used to assist users in reducing their exposure to hate speech online, by researching existing solutions and surveying members of the public on a proof of concept solution.

1.3 Research Question

This research project aims to answer the following questions. These questions ensure that focus was kept on relevant areas of research, as well as being able to recall them at the end of the project to evaluate how they were met.

1. What current technologies exist to prevent hate speech online?

This question will be approached using a literature review where academic research will be evaluated as well as examining real life examples.

2. What might a solution to blocking hate speech look like?

This question will be answered with the development of a proof of concept browser extension.

3. How does the general public feel about that solution?

This question will be answered by presenting a survey to members of the public, with a demonstration of a proof of concept extension. These results will then be analysed.

1.4 Scope

This paper will involve research regarding methods of preventing hate speech online using ML and will include discussion on NLP techniques and evaluation of these techniques. As well as this, research will be carried out into the appropriate methodology to conduct a survey on a proof of concept Hate Speech blocker.

Although a definition of hate speech is discussed and defined for the purpose of this paper, the validity of different definitions won't be discussed. Finally, this paper won't touch on different philosophical arguments for or against censorship of hateful content online.

1.5 Structure

Section 2 of this paper will involve a literature review into defining hate speech, as well as how existing technologies used to manage hate speech. Machine learning will also be explored and how it can be used to detect sentiment. Section 3 will document the development of a proof of concept browser extension and server, detailing design proposals as well as different methods considered in the development of the tool. Section 4 will document the methodology used to develop a survey that was published to the public, along with documenting which information would be collected from participants. It also covers the development of a video demonstration users would view to assess the extension. Section 5 will cover the results of the development of the proof of concept extension, demonstrating how the final extension functions. As well as this, survey results will be presented. Section 6 will go into discussion of these results, with reflection on the execution of the practical aspects of the project, as well as analysis of survey results and comparing quantitative and qualitative results from participants. Finally, Section 7 will summarise the project as a whole, as well as presenting current limitations and suggestions for future work.

2 Literature Review

2.1 Introduction

The following literature review was carried out to review how browser extensions as well as different technologies such as ML can assist in accurately detecting instances of hate speech online. It will also discuss how this effectiveness can be measured, how to notify users of this speech, and issues regarding building datasets as well as reviewing existing tools. This area of research has been of significant interest recently due to the rapid improvement of ML and language recognition, as well as public discussion about censorship and how much or little social media companies should censor content on the platforms. Previous work has been carried out researching how training data can be selected and used to build a model, as well as how the accuracy of such a model can be assessed. As well as this, literature was reviewed regarding the effectiveness of giving users control of the content they're exposed to. The literature review was completed utilising the Abertay Digital Library as well as search engines such as Google Scholar and Science Direct.

2.2 Defining Hate Speech

The term "Hate speech" does not have a universal definition, and the scope of the term can depend on which definition is used. The majority of developed democratic countries have laws defining and restricting the use of it to differing extents (Howard, 2019), with the United Kingdom defining Hate Speech as an expression of hatred towards someone on account of that person's "colour, race, disability, nationality, ethnic or national origin, religion, gender identity, or sexual orientation". Additionally, "Any communication which is threatening or abusive, and is intended to harass, alarm, or distress someone" is illegal as of the 1994 Criminal Justice and Public Order Act (The Crown Prosecution Service, 2017).

Although the United States doesn't have hate speech written into law due to Supreme Court rulings around the First Amendment (*Matal v. Tam*, 2017), most major social media platforms define hate speech in similar terms as the United Kingdom. Facebook, the largest social media platform globally, defines hate speech as a direct attack on someone's protected characteristics, including “race, ethnicity, national origin, religious affiliation, sexual orientation, sex, gender or gender identity, or serious disabilities or diseases” (Facebook, 2021a).

It has been found that regular exposure to this content can be harmful to the individual or groups that are directly targeted, with (Leets, 2002) concluding in a paper, examining the effects of anti-Semitic and homophobic hate speech, that exposure can cause heightened stress, anxiety, depression and desensitisation. This conclusion is supported by a study run by the Economist Intelligence Unit (EIU, 2020) which finds that 65% of women experience hate speech online.

2.3 Technology to Manage Hate Speech

As hate speech online has moved more to the forefront of public discussion, in part thanks to public campaigning and the growing unanimity of social media, tools have been developed to try and mitigate this. Research has been carried out in using techniques such as keyword detection, account blacklisting, and ML based approaches.

The Negator tool (Jain and Kamthania, 2020) makes use of NLP with an Aspect-based Sentiment Analysis (ABSA) model. The extension uses server-side processing to detect the intensity of the speech from 0 to 100%, taking into consideration the topic and who/what the speech is aimed at, then categorising the speech into topics including “Abuse, Personal Attacks, Cyberbullying, Sexual Advances, Bigotry, Criminal Activity” and “Death Threats”, and will block the post with a visual indicator notifying the user which category the speech falls under if the intensity is more than 60%.

The Shinigami Eyes tool takes a different approach in notifying users of harmful content. The extension focuses on transphobic social media accounts and websites, with users submitting links as being “anti-trans” or “trans-friendly” that are manually reviewed before being added to the list, implemented using a bloom filter.

Modha *et al.* (2020) proposes a browser extension that implements the TRAC dataset, which classifies training data into sentiment categories of “Overtly aggressive”, “Covertly aggressive” and “Non-aggressive”, showing users the levels of each category embedded in the web page.

2.3.1 Displaying & notifying users of hateful content

There are numerous different proposed methods to alert the user of hateful content, ranging from blocking the post from view completely to just giving the user a notice that the account has a history of hateful conduct. The Negator tool mentioned in *Existing Tools* takes a harsh stance on hiding hateful content and entirely hides posts that meet its own criteria with an interface notifying the user that the content has been blocked, giving the category the post falls into, with an option to view the post anyway. This is shown below in Figure 1.

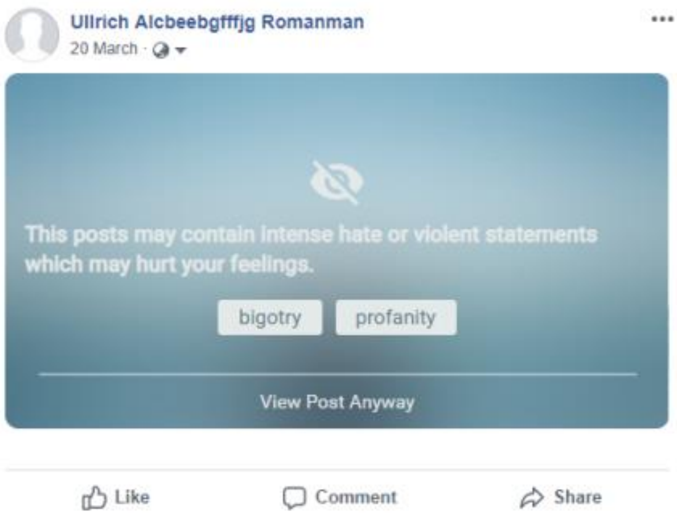

Ullrich Alcbeebgfffjg Romanman
20 March ·
This posts may contain intense hate or violent statements which may hurt your feelings.
bigotry
profanity
View Post Anyway
Like
Comment
Share

Figure 1 - Negator interface on detection of hateful content

Shinigami Eyes uses colour coded warnings marking hyperlinks and profiles as red (anti-trans) or green (trans-friendly) as shown in Figure 2. This approach is less harsh in that it still shows users the offending user's posts but can give extra context that might increase the likelihood of users not interacting with "anti-trans" accounts. Users may submit reports via the extension interface by right clicking on a link to a profile or website.

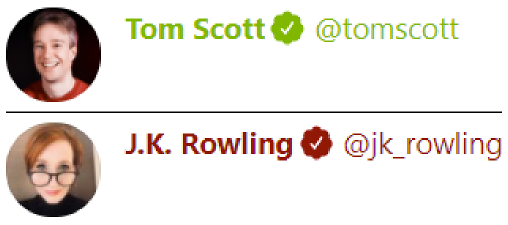
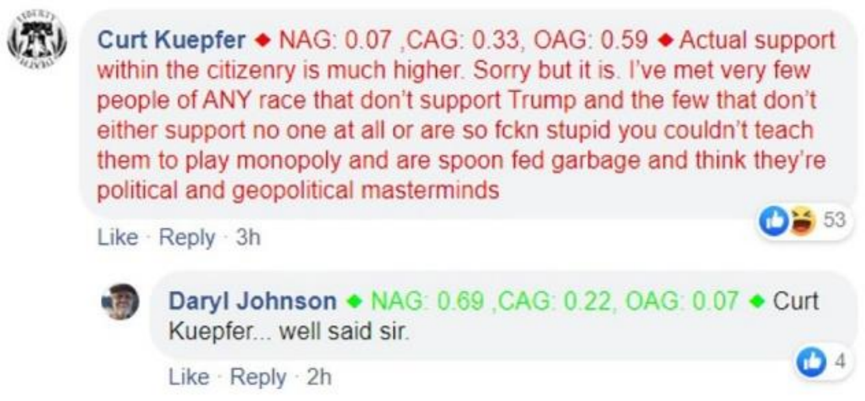


Figure 2 - Shinigami Eyes extension presenting a "trans-friendly" account in green and an "anti-trans" account in red

The extension demonstrated by Modha *et al.* (2020) uses a similar method of colour coding as the Shinigami Eyes extension, with posts that pass a high threshold of confidence displayed completely in red, with medium displayed in yellow and non-hateful posts in green. Along with this, the levels detected by the model are shown directly above the comment as a number. The decision was justified in the paper by the distrust that the public has regarding algorithms and ML, partly due to the fact they are seen as a black box. The extension's output is shown in Figure 3.



Curt Kuepfer ◆ NAG: 0.07 ,CAG: 0.33, OAG: 0.59 ◆ Actual support within the citizenry is much higher. Sorry but it is. I've met very few people of ANY race that don't support Trump and the few that don't either support no one at all or are so fckn stupid you couldn't teach them to play monopoly and are spoon fed garbage and think they're political and geopolitical masterminds

Like · Reply · 3h 53

Daryl Johnson ◆ NAG: 0.69 ,CAG: 0.22, OAG: 0.07 ◆ Curt Kuepfer... well said sir.

Like · Reply · 2h 4

Figure 3 - Modha et al. browser extension on a hateful comment showing levels of Non-aggression, Covert-aggression, and Overt-aggression.

2.4 Machine Learning

2.4.1 Client side & cloud models

As ML has gained popularity in business applications, commercial services are now common where users can rent computing power and take advantage of mature pre-existing models. Lee, (2018) investigates this in a paper analysing the current state of the industry. AWS SageMaker, Microsoft Azure ML Studio, Google Cloud ML Engine and BigML are found as the major players in this space, with Google offering Python APIs for voice, NLP, translation, and other features. Robust libraries and frameworks exist for a multitude of languages for building a model locally, with Python offering libraries such as scikit-learn, mlxtend and Shogun that can be configured with most ML algorithms involving transformation, decision trees, classification and more (Stančin and Jović, 2019).

2.4.2 Datasets

A significant factor in training an accurate ML model is to ensure there is a well annotated dataset that is relevant to the topic being worked on. In the area of linguistics, the term “corpus” refers to a collection of texts, especially regarding a particular subject. In the field of NLP, a corpus similarly refers to a structured set of text or data, commonly used for training a model. However, classifying hate speech has proven to be a challenge, principally due to the loose definition of the term.

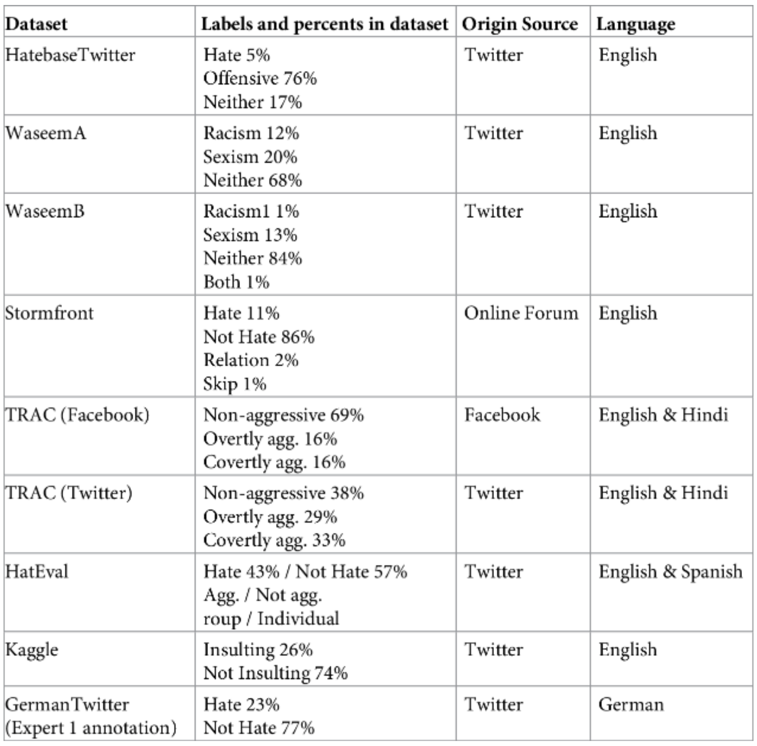
Ross *et al.* (2016) discusses this in the context of training a dataset from the online discourse surrounding the European Refugee Crisis, with data gathered from 13,766 Tweets including a selection of offensive hashtags. The Tweets were categorised by annotators who were also given Twitter’s official definition of hate speech, but the rate of agreement was still low with a Krippendorff's alpha of $(\alpha) = 0.38$ (with 0 being complete disagreement and 1 being complete agreement).

The paper describes the significant hateful sentiment seen on social media aimed at refugees, and it is concluded that there needs to be a stronger and more consistent definition of hate speech. Additionally, when annotating datasets, finding common characteristics of content users find hateful will be useful in building a more automated detection model.

MacAvaney *et al.* (2019) discusses gathering data in a paper focusing more widely on the challenges and possible solutions of hate speech detection. An issue brought up is the data usage and distribution policies of major social media companies who want to restrict users scraping their platform for various reasons including legitimate user privacy concerns. This creates an issue on Facebook, the biggest social platform and location for a significant amount of online discourse, not allowing for scraping of content. Therefore, a useful stream of data is restricted – for example discussion in comments underneath a controversial news article.

There are still methods to gather this data, however it requires the use of third-party tools, and it is against Facebook’s Terms of Service. Most hate speech datasets therefore are gathered from platforms such as Twitter, due to its more relaxed policy and open APIs, as well as more niche and directly hateful websites. The paper also collates a list of sources for hate speech focused text datasets, as shown in Table 1.

Table 1 - List of open source hate speech datasets collated by MacAvaney et al., (2019)



Dataset	Labels and percents in dataset	Origin Source	Language
HatebaseTwitter	Hate 5% Offensive 76% Neither 17%	Twitter	English
WaseemA	Racism 12% Sexism 20% Neither 68%	Twitter	English
WaseemB	Racism1 1% Sexism 13% Neither 84% Both 1%	Twitter	English
Stormfront	Hate 11% Not Hate 86% Relation 2% Skip 1%	Online Forum	English
TRAC (Facebook)	Non-aggressive 69% Overtly agg. 16% Covertly agg. 16%	Facebook	English & Hindi
TRAC (Twitter)	Non-aggressive 38% Overtly agg. 29% Covertly agg. 33%	Twitter	English & Hindi
HatEval	Hate 43% / Not Hate 57% Agg. / Not agg. roup / Individual	Twitter	English & Spanish
Kaggle	Insulting 26% Not Insulting 74%	Twitter	English
GermanTwitter (Expert 1 annotation)	Hate 23% Not Hate 77%	Twitter	German

A related issue is the disproportionate language representation in hate speech datasets, with English making up the vast majority due to being the go-to language for online discussion. For example, MacAvaney et al., (2019) discusses how the lack of a German language hate speech corpus necessitated the construction of their own corpus.

Another example of this is the Stormfront dataset (Fišer *et al.*, 2018), a collection of thousands of sentences collected from one of the longest running neo-Nazi websites in existence. These sentences were manually annotated, marked as either having hate speech or not. Although a valuable resource, any non-English posts were filtered out using an automated language detector. This is an issue more widely in the field as building a model in a language less widely used online could require manually building a dataset, which can be a time-consuming task.

Mubarak, Darwish and Magdy, (2017) address this in a paper that covers the gathering of an Arabic dataset as well as building a model. Data was gathered via Twitter using the API's language filter "lang:ar". As well as this, data was gathered from the regionally popular Al Jazeera news site, with 32k deleted comments. This encompasses hateful content along with comments not relevant to the article's topic and spam/advertising.

2.4.3 Assessing accuracy

There are various methods to assess the accuracy of natural language models, and the appropriate measurement can vary depending on what the NLP model is intended to be used for. A reliable method available with certain datasets is making use of “Dev set” data.

This was implemented as part of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-1) (Kumar *et al.*, 2018), where 130 teams helped annotate 15,000 Facebook comments and posts. This large data pool meant a second non-overlapping corpus could be created, which came from similar sources to the intended training data making it a good testing environment. Once test data is acquired, a popular metric to describe the performance of ML models is a Confusion matrix.

Malmasi and Zampieri, (2017) make use of this in a paper making use of the Hate Speech Detection dataset using a Support Vector Machine (SVM) classifier, using a confusion matrix to show what the model struggles with the most as shown in Figure 4.

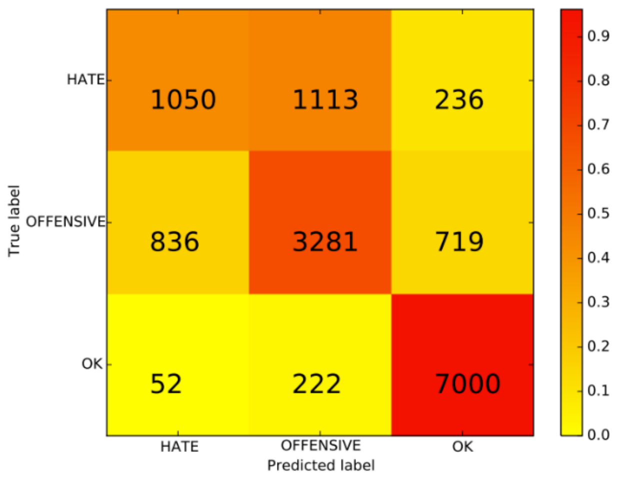


Figure 4 - Confusion matrix from Malmasi and Zampieri (2017) measuring model accuracy

The following formulas shown in Figure 5 show how the false positive (FP) and false negative (FN) rates are found, with TP representing results accurately recorded as true and TN representing correctly recorded negative results. The rate can range from 0-1, with 0 representing 100% accuracy and 1 representing 0% accuracy.

$$False\ Positive\ rate = \frac{FP}{FP + TN}$$

$$False\ Negative\ rate = \frac{FN}{FN + TP}$$

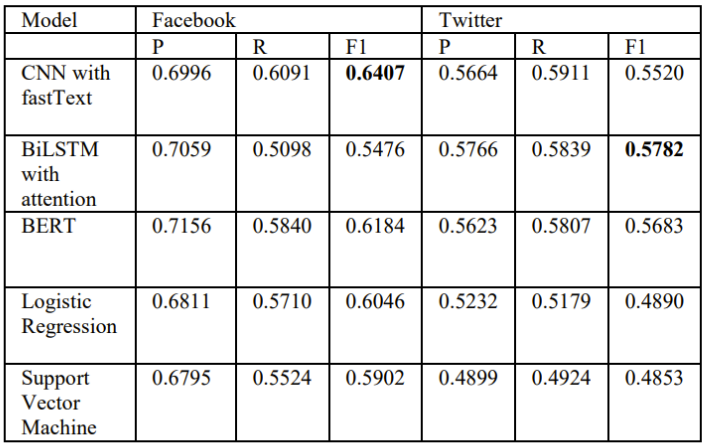
Figure 5 - Formulas to find False Positive & False Negative rates

2.4.4 Comparing deep learning models

Many different models and techniques exist that enable Deep Learning in NLP. These models use different technologies that result in different levels of accuracy. Kapil, Ekbal and Das, (2020) investigate this, comparing popular models including “Convolutional Neural Network” (CNN), “Character-CNN”, “Long short-term memory” (LSTM) and “Bi-directional LSTM” (BiLSTM), with multiple publicly available datasets. It was found that LSTM and BiLSTM performed the best for all datasets.

Modha *et al.*, (2020) performed a similar benchmark using the TRAC-1 dataset, comparing results between other popular models. As shown in Table 2, “CNN with fastText” was the most accurate on English Facebook, and “BiLSTM with attention” was more accurate on English Twitter.

Table 2 - Test results based on TRAC-1 dataset, P = precision, R = Recall, F1 = weighted F1 score



Model	Facebook			Twitter		
	P	R	F1	P	R	F1
CNN with fastText	0.6996	0.6091	**0.6407**	0.5664	0.5911	0.5520
BiLSTM with attention	0.7059	0.5098	0.5476	0.5766	0.5839	**0.5782**
BERT	0.7156	0.5840	0.6184	0.5623	0.5807	0.5683
Logistic Regression	0.6811	0.5710	0.6046	0.5232	0.5179	0.4890
Support Vector Machine	0.6795	0.5524	0.5902	0.4899	0.4924	0.4853

2.5 Summary

The literature reviewed provided a significant insight into the premise for using deep learning methods in helping prevent hate speech online. Existing work in the area shows the potential applications a browser extension could have in the real world, with different methods effective at helping users control their exposure to hateful content and be informed of who they are interacting with. The papers reviewed include detailed discussion of the different natural language models for detection but conclude that each method has its drawbacks and can all perform well with high quality training data. The research also provided valuable insights into datasets, highlighting the value of properly sanitised and annotated data to help train and test language models.

3 Prototype Development

This section will cover how the technical side of the project was designed and implemented. This task was split into two distinct parts – developing a Chrome browser extension and a Python application, with the browser extension used to scan webpages for elements, and a Python server to receive these elements and process them, sending back the results. Research was carried out into the process of developing a browser extension, as well as appropriate ways of processing the data required.

3.1 Project Planning

An iterative project planning method was used to carry out this project. This involved using a Gantt chart and breaking down tasks into increments, first implementing a base for the browser extension, then extending into word detection functionality, server-side processing etc. This project planning method as shown in Figure 6, also allowed for iterating individual parts of the project after developing, with the browser extension going through multiple functional versions.

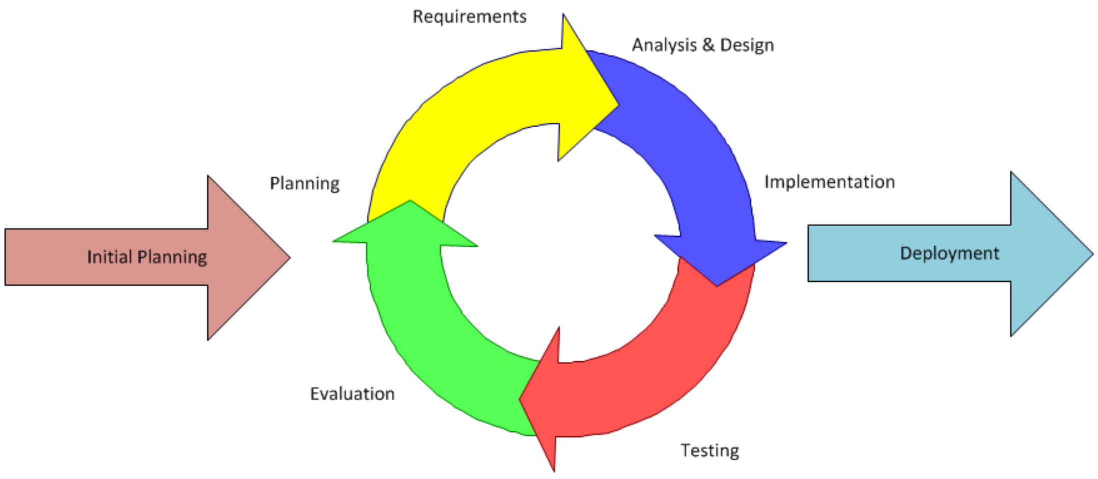


Figure 6 - Demonstration of Iterative project planning method (Inflectra, 2021)

A risk assessment was carried out on the viability of the development of the project, considering factors such as stages taking longer than estimated or bugs being found during the testing process. As shown in Appendix B – Risk Assessment, Risk 1-5 relate to the project being in an area that's relatively unfamiliar to the researcher, involving ML and browser extension development. This risk was mitigated by allocating more time than was estimated to be needed in these steps, giving time to revise basics if needed. Risk 6 relates to finding bugs in the extension's code during the development process, which is extremely likely. Overall, this was perceived as a low risk, both due to the generous amount of time allocated to learning, and the fact that the languages being implemented (JavaScript and possibly Python) are interpreted so therefore don't need to be compiled in order to test, only requiring restarting the application to apply changes. Other risks relate to personal factors such as working remotely and stress caused by other coursework, which can be mitigated by maintaining a schedule and keeping in contact with classmates and the project supervisor.

3.2 Design

An initial design was proposed for the extension during the planning phase. This prototype was designed in the online diagram software Draw.io (Diagrams.net, 2020). The overall design was aimed at being accessible and self-explanatory, with a red "HS" icon in the browser task bar (abbreviated "Hate Speech") and pop-up interface notifying if hateful content is present, along with some extra information on the website and an indicator showing whether the extension is active or not. The design proposal is shown in Figure 7.

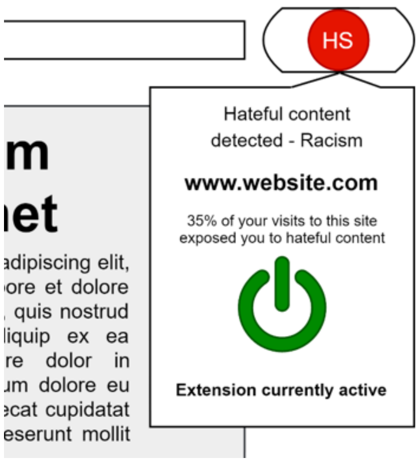

HS
Hateful content detected - Racism
www.website.com
35% of your visits to this site exposed you to hateful content
Extension currently active

Figure 7 - Prototype design of extension pop-up

A plain white background and black font was decided on as according to Hall and Hanna, (2004), "For educational sites, where retention and readability are a major concern; black on white or a closely related combination of text should be used". As well as this, it leads to the extension blending in with most text-based sites where the standard is light background and dark text.

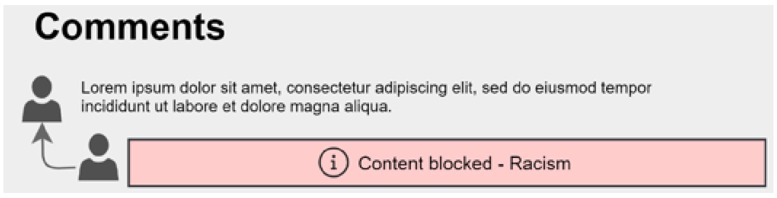

Comments
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Content blocked - Racism

Figure 8 - Prototype design of extension overlay

A proposed design for how the extension will hide content was also made. This design was intended to make it clear to the user which posts were hidden at a glance, with an informational notice advising why it was blocked. A red background was chosen as it has been observed using colour psychology to catch attention in an emotional context (Kuniecki, Pilarczyk and Wichary, 2015).

3.3 Development

The browser extension was developed for the Google Chrome browser. This decision was made as Chrome is currently the market leader in desktop web browsers, with 67% of market share worldwide as of March 2021 (StatCounter, 2021). This made it more likely participants would be familiar with how this browser functioned. As well as this, the Chromium framework is used in other browsers such as Microsoft's Edge browser as well as the Opera browser meaning the extension would run in either. Additionally, as browser extensions are developed primarily using cross platform web technologies like JavaScript, it would be trivial to port over to a non-Chromium browser such as Mozilla Firefox.

Originally, the browser extension was designed to deal with both detecting and processing the webpage's data, however during the development process it was found that this increased the complexity of the code significantly, along with making it more difficult to swap out the processing algorithm later down the line. Therefore, the decision was made to split the application into 2 distinct parts as shown in Figure 9 – the Chrome web extension, to parse the webpage elements and send & receive this data, and a separate Python based server which retrieves this data, processes it and sends it back to the web extension.

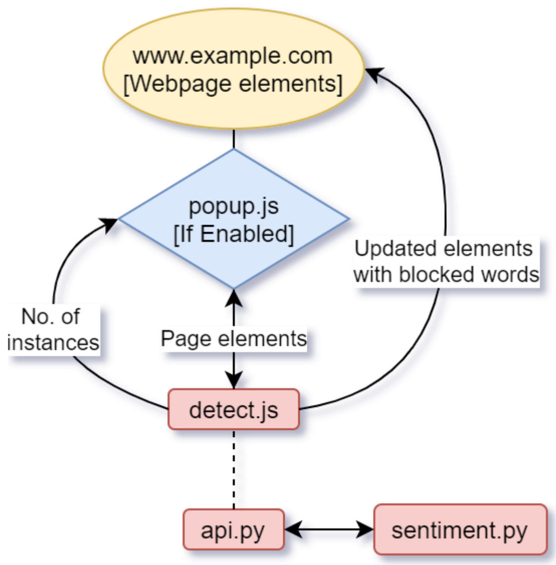
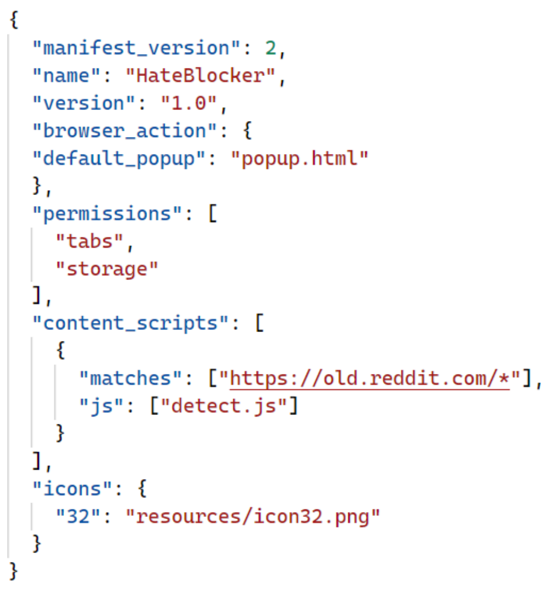


Figure 9 - Flow diagram of application

3.3.1 Text detection

In order for the browser extension to detect and block hateful content on webpages, it had to be able to read elements on the page, as well as being able to filter which elements would be relevant to the search. In order for browser extensions to inject scripts into webpages, `content_scripts` had to be declared through `manifest.json` as shown in Figure 10. This file is required in order for the extension to run, as it holds the extension's metadata and other configuration.

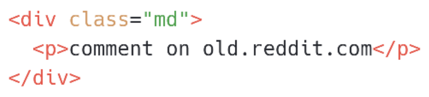


```
{
  "manifest_version": 2,
  "name": "HateBlocker",
  "version": "1.0",
  "browser_action": {
  "default_popup": "popup.html"
  },
  "permissions": [
    "tabs",
    "storage"
  ],
  "content_scripts": [
    {
      "matches": ["https://old.reddit.com/*"],
      "js": ["detect.js"]
    }
  ],
  "icons": {
    "32": "resources/icon32.png"
  }
}
```

Figure 10 - Manifest file of HateBlocker

The technique used for fetching relevant elements depends on which site the HateBlocker is being aimed at, for example as seen in Figure 10 the content script is being aimed at `old.reddit.com`, this being the classic interface of the popular Reddit social media platform. Reddit was chosen for the proof of concept version of HateBlocker as since it's a decentralized platform, relying on voluntary moderators to manage their own communities, it's been widely reported that some communities don't strictly enforce Reddit's code of conduct and leave hateful content on the site (Newcomer, 2020).

The classic interface was chosen specifically as it doesn't rely on much JavaScript to draw interface elements and therefore the Document Object Model (DOM) is relatively static and easy to parse. As shown in Figure 11, the text from each post is wrapped in a `<p>` HTML paragraph element so this was used as a starting point to fetch from.



```
<div class="md">
  <p>comment on old.reddit.com</p>
</div>
```

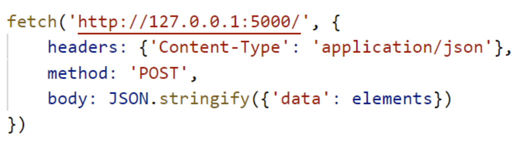
Figure 11 - <p> element on Reddit that was targeted

This method could easily be adapted to forums and imageboards such as 4chan, however a different method would need to be adopted for modern social media platforms such as Twitter, Facebook, and the more recent design of Reddit as they use the popular ReactJS library (Facebook, 2021b). This library is used to build user interfaces and increases the complexity of detecting elements on a webpage as unlike static webpages that write directly to the DOM, to improve performance React virtualises the DOM, only periodically updating the real version.

As well as this, react allows for implementing features such as infinite scrolling, which wouldn't be picked up by this proof of concept as it fetches elements when the page is first loaded. Finally, after fetching the relevant elements, the `innerText` of these elements was added to an array. This removed parts of the element object that were unneeded for processing the text.

3.3.2 API

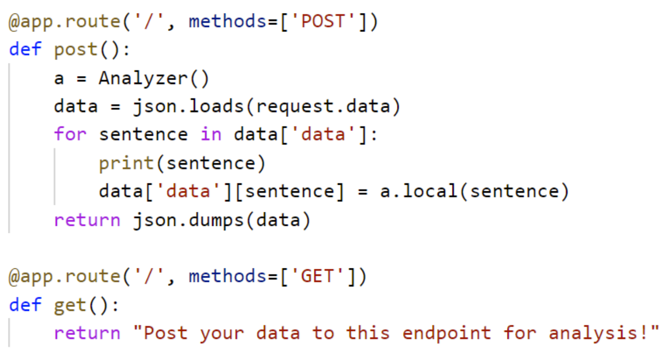
The development of the project was split into two parts, with the extension sending webpage elements via POST request as JSON data - a standardized format for serializing data, in real time to a remote server. As shown in Figure 12, this is done asynchronously using the `fetch` API available in JavaScript ES6.



```
fetch('http://127.0.0.1:5000/', {
    headers: {'Content-Type': 'application/json'},
    method: 'POST',
    body: JSON.stringify({'data': elements})
})
```

Figure 12 - Extension code that posts data to server

This data is then received by a Python Flask (Flask, 2020) application located at `app.py`. Flask was chosen as it's a lightweight web framework that allowed for a POST endpoint to be configured in a few lines of code. As shown in Figure 13, two methods were added to the server, to receive POST requests and send this data to the analyser, and to send a response if a GET request is mistakenly sent to the server.



```
@app.route('/', methods=['POST'])
def post():
    a = Analyzer()
    data = json.loads(request.data)
    for sentence in data['data']:
        print(sentence)
        data['data'][sentence] = a.local(sentence)
    return json.dumps(data)

@app.route('/', methods=['GET'])
def get():
    return "Post your data to this endpoint for analysis!"
```

Figure 13 - Flask app endpoints

Once data was passed into the Flask application and processed by the analyser, it was returned to the extension via the POST response, formatted as JSON data, with an extra property of either True, meaning the text was detected to be hateful, or False meaning the text didn't trigger the analyser. Shown in Figure 14 is the flow of how the endpoints communicate with the extension. This API based method of communication between the extension and server makes it easy for future work in to be carried out in implementing different extensions or servers.

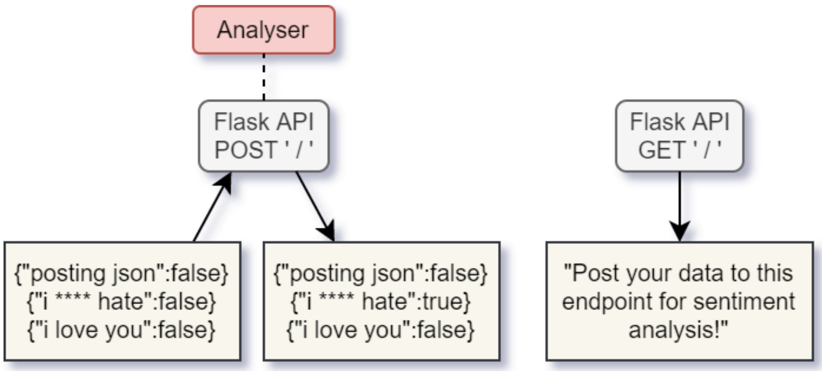
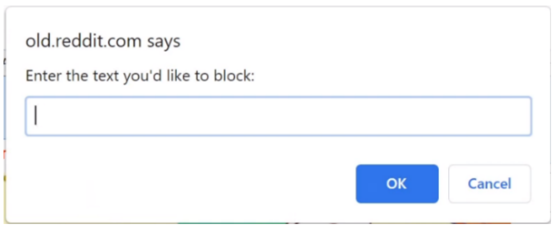

Analyser
Flask API POST '/'
Flask API GET '/'
{"posting json":false} {"i **** hate":false} {"i love you":false}
{"posting json":false} {"i **** hate":true} {"i love you":false}
"Post your data to this endpoint for sentiment analysis!"

Figure 14 - Diagram visualising API flow

3.3.3 Text processing

Text processing functionality was developed with the goal in mind of being interchangeable, allowing for different hate speech detection techniques and algorithms to be added further down the line. A local version of text processing was implemented in JavaScript in the Chrome extension earlier in the development stage, and this version was demonstrated in the survey video.

Different avenues were explored in this area, one of which being an extension of the original word detection method – Instead of a single word being entered in an alert box as shown in Figure 15, the text elements would be processed on the server side using a wordlist. Other methods were explored involving sentiment analysis using Google's Cloud Language and Perspective APIs.


old.reddit.com says

Enter the text you'd like to block:

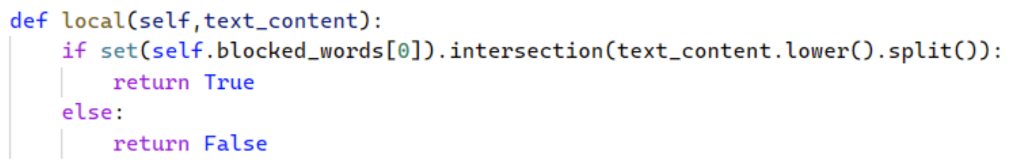
Figure 15 - Original UI for selecting word to block

Since Python was used to implement the API, one programmed, it was implemented by importing `analyzer` into the Flask application as a library.

3.3.3.1 Local wordlist

Shown in the above section, a local wordlist was the first step in detecting overtly hateful text online. This method misses out on the context in which the term was used - for example, if the word is offensive only in the context of a particular discussion. The issue is discussed further in the Discussion section. Although relatively easy to implement technically, a wordlist had to be found or generated for the Python application to use.

Hatebase (*Hatebase*, 2021) provides a comprehensive and regularly updated list of hateful words in use in multiple languages globally, and was considered for use in the application. The site requires use of an API to access their dataset which could considerably slow down the runtime of the application, however another suitable dataset was found courtesy of a research group from (Carnegie Mellon University, no date) listing 1,300+ English terms that could be found offensive.



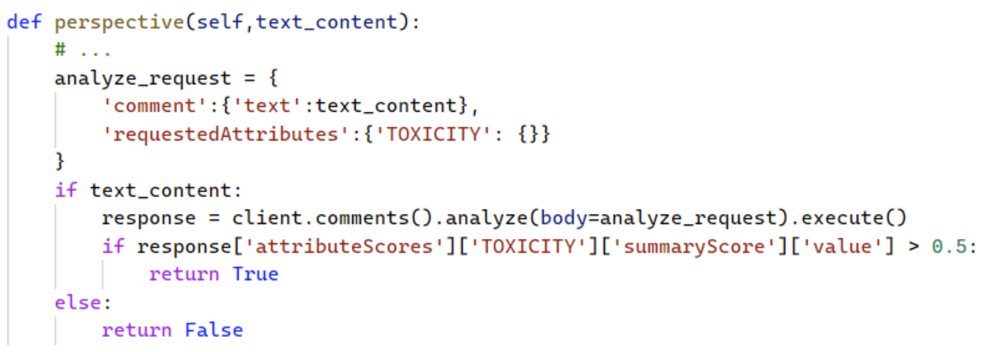
```
def local(self,text_content):
    if set(self.blocked_words[0]).intersection(text_content.lower().split()):
        return True
    else:
        return False
```

Figure 16 - Local wordlist checking for matches with webpage text

This dataset was downloaded as a TXT file, and was converted to CSV using Microsoft Excel, converting new lines in the file to commas. As shown in Figure 16, `text_content` was converted to a lower case list, then checked against `blocked_words` for any intersections, which would happen if the word was present in both lists, causing the Function to return True and therefore the element being marked as hateful.

3.3.3.2 Cloud based sentiment analysis

After implementing word detection using a local Python server, cloud-based sentiment analysis was integrated into the project as a demo of the extensibility of a server-based approach. This was done using Google's Perspective AI. This is a limited access API developed by Jigsaw - a research unit within Google and Google's Counter Abuse Technology teams, to enable better conversations online by creating ML models targeted at online toxicity and harassment (Perspective Developers, 2019). After applying for access, it was added to a Google Cloud Console project and accessed via API keys.



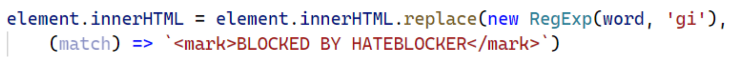
```python
def perspective(self,text_content):
    # ...
    analyze_request = {
        'comment':{'text':text_content},
        'requestedAttributes':{'TOXICITY': {}}
    }
    if text_content:
        response = client.comments().analyze(body=analyze_request).execute()
        if response['attributeScores']['TOXICITY']['summaryScore']['value'] > 0.5:
            return True
    else:
        return False
```

Figure 17 - Perspective API implementation

As shown in Figure 17, the JSON response was parsed to get the toxicity score for the paragraph, measured between 0 (no toxicity) and 1 (extremely toxic), with any value over 0.5 returning True resulting in the element being marked as hateful.

3.3.4 Notifying the user

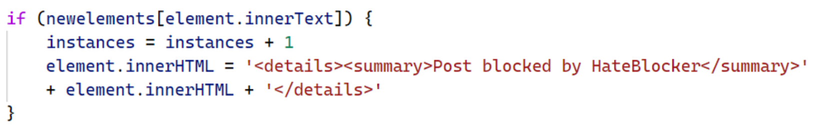
Due to how elements are sent to a server and returned, limitations were found regarding how the elements could be modified, to notify the user that they had been blocked. Initially, elements were checked to see if they included the blocked word, and if they did, the `innerHTML` was appended to remove the phrase and replace it with "BLOCKED BY HATEBLOCKER", highlighted using the `<mark>` tag shown in Figure 18. As demonstrated, only HTML can be appended using this method, which limits the visual appearance of the text.



```javascript
element.innerHTML = element.innerHTML.replace(new RegExp(word, 'gi'),
    (match) => `<mark>BLOCKED BY HATEBLOCKER</mark>`)
```

Figure 18 - Initial method of replacing element text

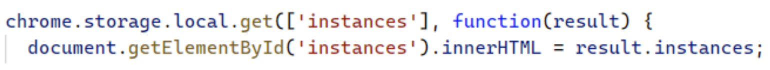
Another method which was used for the final version of the extension had the same limitation but was modified, so users can choose to show/hide the blocked text using a dropdown menu implemented by using the HTML `<details>` tag as shown in Figure 19.



```javascript
if (newelements[element.innerText]) {
    instances = instances + 1
    element.innerHTML = '<details><summary>Post blocked by HateBlocker</summary>'
    + element.innerHTML + '</details>'
}
```

Figure 19 - Final method of replacing element text

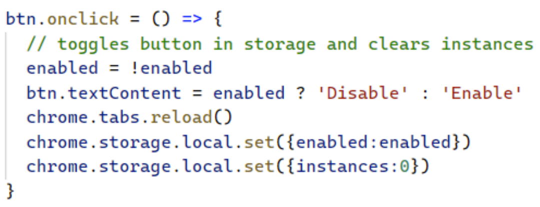
As well as notifying the user by covering offending elements, the extension's pop-up was configured to show the number of instances on a given page. This was done by adding a temporary counter to Chrome’s local storage API after instances are returned, which is fetched when the pop-up is opened as shown in Figure 20.



```
chrome.storage.local.get(['instances'], function(result) {
  document.getElementById('instances').innerHTML = result.instances;
```

Figure 20 - Fetching instances from storage API

This result was cleared on page change, as well as when the extension was disabled using the pop-up interface as shown in Figure 21.



```
btn.onclick = () => {
  // toggles button in storage and clears instances
  enabled = !enabled
  btn.textContent = enabled ? 'Disable' : 'Enable'
  chrome.tabs.reload()
  chrome.storage.local.set({enabled:enabled})
  chrome.storage.local.set({instances:0})
}
```

Figure 21 - Code run when Enable/Disable button pressed

4 Research Methodology

In order to evaluate the usefulness of the extension and gather opinions from the public, research was carried out on appropriate methods of measuring this. The two formats most considered were in-depth interviews and user acceptance surveys. An interview would involve gathering volunteers for a 1-on-1 meeting with the participant, where the extension would be demonstrated live by the researcher or the participant would use the extension directly with guidance from the researcher. Questions could be pre-prepared for the participant to answer, with an advantage being that the participant can ask follow-up questions to better inform their answers, as well as giving any feedback they might have while using the extension.

An explanatory-sequential approach was decided on as the design methodology for surveying participants. This involved gathering quantitative data as it allows aggregate results to be expressed in numbers, graphs, and measurements. Qualitative data was also collected from participants to help explain the human reasoning behind the qualitative results (Edmonds and Kennedy, 2017). Following this methodology would require finding an appropriate metric for surveying both types of results.

A user acceptance survey was conducted, both to gather opinions on the proposed hate speech blocker browser extension, as well as how this relates to different demographic information including age, gender, and level of comfort of exposure to online hate speech currently. This was decided on instead of another form of evaluation such as an interview, which focuses on gathering in-depth opinions from a small group of people, as it allowed the researcher to gather a large amount of aggregated data from diverse groups of people that wouldn’t always be accessible via interview.

Due to the COVID-19 pandemic, participants were unable to easily test the browser extension in person. Although options were considered with regards to allowing users to install the extension on their personal machines, due to the extension still being in development it would require extracting a ZIP file and enabling a developer mode in Google Chrome. The decision was made that this would be too awkward for novice users as well as being a security risk.

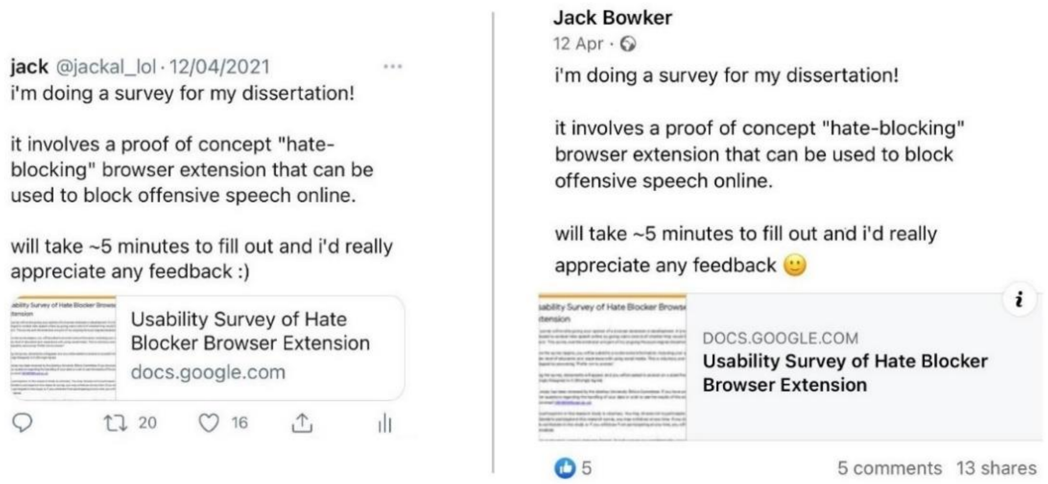
A risk assessment was carried out to assess the viability of the surveying stage of the project, similarly to as shown in Section 1.1 – Development Risk Assessment. As shown in Appendix

B – Risk Assessment, the main risk relating to the surveying process is gathering suitable participants. This was marked as a high risk, due to the relative uncertainty of knowing how many members of the public would be willing to install an unknown browser extension and fill out a survey. This risk was mitigated by planning in advance to estimate how many participants to be aimed for, and who the survey would be sent out to. As well as this, the requirement of installing a browser extension was dropped in exchange for watching a short demonstration video.

The survey was published on the 12th of April 2021 giving time to analyse results and modify the extension based on any feedback given.

4.1 Recruitment and Ethics

This survey was focused on social media users, aiming for a proportionate representation of age groups and gender. Participants were recruited using a combination of social media and e-mail. A Google Forms link was posted publicly to personal social media platforms (Facebook, Twitter, Instagram, and LinkedIn) and sharing was encouraged, as well as being sent out via e-mail to friends and family.


jack @jackal_lol · 12/04/2021
i'm doing a survey for my dissertation!

it involves a proof of concept "hate-blocking" browser extension that can be used to block offensive speech online.

will take ~5 minutes to fill out and i'd really appreciate any feedback :)

Usability Survey of Hate Blocker Browser Extension
docs.google.com

20 16

Jack Bowker
12 Apr ·
i'm doing a survey for my dissertation!

it involves a proof of concept "hate-blocking" browser extension that can be used to block offensive speech online.

will take ~5 minutes to fill out and i'd really appreciate any feedback 🙂

DOCS.GOOGLE.COM
Usability Survey of Hate Blocker Browser Extension

5 5 comments 13 shares

Figure 22 - Social media posts recruiting participants on Twitter and Facebook

The survey was reviewed by the Abertay University Ethics Committee, as originally the demonstration was to include exposure to hateful comments & posts directly. Permission for this was given but not needed, as the video demonstration replaced hateful words with harmless examples. In order to fulfil the ethical commitments agreed to, before giving any information, participants were informed of what data they would be offered to give, and how this data was going to be used. They were also informed their data could never be tied back to them individually, and of their right to withdraw their consent at any time, by either not submitting the survey or by contacting the researcher directly. Finally, a GDPR compliance form was completed and is viewable in Appendix C – GDPR Sign Off Form.

4.2 Survey Instrument

The survey was implemented using the online platform Google Forms (Google, 2014). This platform was chosen due to its ease of access along with integrating well with YouTube, allowing for embedding the video directly on the survey page.

4.2.1 Demographic Information

Before filling out the survey, participants were given the choice to provide additional demographic information that would be linked to their answers provided in the survey. Specifically, age group, gender, and level of education were asked for, along with asking users to enter the social media platforms they use by ticking boxes. These were requested to analyse trends in opinion on the browser extension, and to gain insight into which users may find it most useful. Demographic information requested is shown in Appendix A – Demographic Information. An option was given with every question of ‘Prefer not to say’. This option allowed users that were uncomfortable with giving away demographic information due to privacy concerns etc. to still contribute to the survey.

4.2.2 Video Demonstration

Due to the complicated nature of walking participants through installing a browser extension manually, it was decided that a video would be created to simulate what the browser extension’s experience was like. This involved recording an in-development version of the extension on Reddit, a site the extension was intended for. The recording was done on Windows using Open Broadcaster Software (OBS), a free open-source screen recording application (OBS, 2021).

The demonstration recording involved using the extension in a way that was as easy as possible to understand, which involved planning which areas of the extension to show and when. As shown in Figure 23, a linear diagram was used to assist in visualising the order to carry out the demonstration.

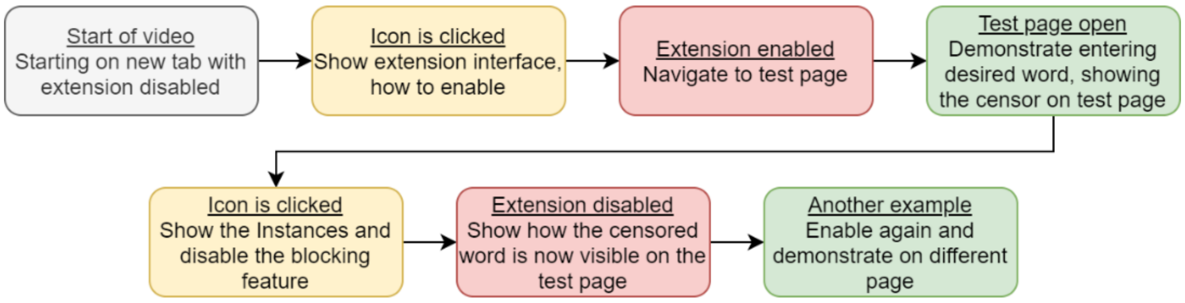

Start of video
Starting on new tab with extension disabled
Icon is clicked
Show extension interface, how to enable
Extension enabled
Navigate to test page
Test page open
Demonstrate entering desired word, showing the censor on test page
Icon is clicked
Show the Instances and disable the blocking feature
Extension disabled
Show how the censored word is now visible on the test page
Another example
Enable again and demonstrate on different page

Figure 23 - Diagram to visualise order of demonstration video

Once the video was recorded, it was edited in Vegas Pro 14 (MAGIX, 2016) to slow down the clip to making it easier to follow. As well as this, annotations were added to commentate each part of the video explaining what was happening. The video was uploaded to YouTube (Jack Bowker, 2021) where it could be embedded into the Google Forms survey.

4.2.3 Quantitative Feedback

The System Usability Scale was used to assess user's subjective rating on the usability of a system or piece of software. It consists of 10 standard questions aimed at measuring the effectiveness, efficiency, and satisfaction of a system/piece of software, scored on a Linear Scale from Strongly Disagree to Strongly Agree. This scale was used as it's an established metric that's been in use since 1996 and it's been proven that it's a highly robust and versatile tool according to Bangor, Kortum and Miller, (2008). These questions were adapted for the browser extension as shown in Figure 24.

Q1 - I think I would use this extension frequently.

Q2 - The extension looked unnecessarily complex.

Q3 - The extension looked easy to use.

Q4 - I think I'd need the support of a technical person to be able to use this extension.

Q5 - I found the functionality of this extension well integrated with the site.

Q6 - The extension was designed in an aesthetically pleasing way.

Q7 - I would imagine that most people would learn to use this extension very quickly.

Q8 - The extension looked very cumbersome to use.

Q9 - I think I would feel very confident using the extension.

Q10 - I would need to learn a lot of things before I could get going with this extension.

Figure 24 – System Usability Scale questions adapted for survey

In order to calculate scores for a System Usability Scale, the scores will be converted to a number (0-4) by subtracting 1 from odd-numbered questions, as these questions are positively framed, and subtracting even-numbered questions from 5, as these questions are negatively framed meaning a smaller score (Strongly Disagree/Disagree) resembles positive feedback.

After this, the result is multiplied by 2.5 resulting in a score of 0-100. This result however shouldn't be viewed a percentage and individual entries should be considered based on their percentile (Affairs, 2013). If viewing the metric as an overview, the average System Usability Scale score is 68, however this can vary based on the complexity of the system being reviewed (Sauro, 2011).

In order to normalize this data, research data was used that involved the results of nearly 1,000 System Usability Scales being analysed, with the goal of determining the percentile of results that the score falls into (Bangor, 2009). As shown in Figure 25, scoring upwards of 71.4% would rank the result as "good", with results of over 85.5% ranked as "excellent".

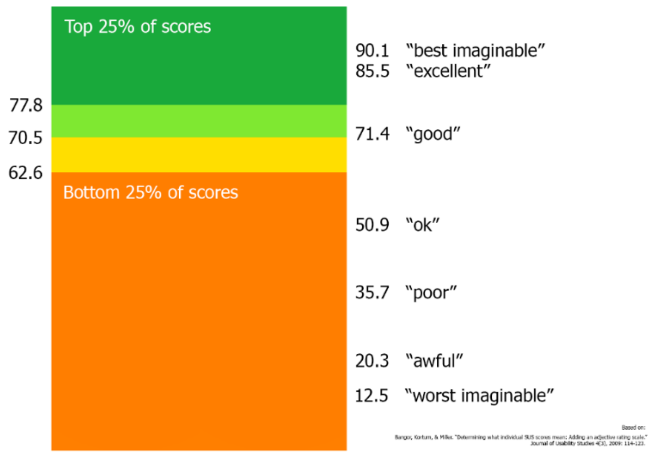


Figure 25 - Graph visualising System Usability Scale percentiles with adjectives

4.2.4 Qualitative Feedback

Along with implementing a System Usability Scale, one additional Linear Scale question ranging from Strongly Disagree – Strongly Agree was added, presenting the assertion that "I'm comfortable with my amount of exposure to hateful content online". This question was added as a factor to determine how useful the extension would be for the user, as well as being an indicator of how the following questions would be answered.

After answering 11 Linear Scale questions, 3 questions were added to the survey that the user could answer by entering text. Firstly, "Is there any other ways the browser extension could be improved, based on the video viewed?". This question was used as a general feedback box for participants to voice their opinion and suggestions on how the extension itself should be executed.

After this, the question "Which websites would you see this extension being useful on?" is asked to allow the user to add site suggestions they would find HateBlocker useful on. This data was collected as if the extension was to be extended in the future, it would be sensible to focus on integrating into the sites most commonly suggested here.

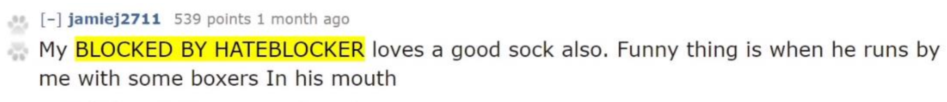
Finally, the question "If uncomfortable using this extension, what would make you more comfortable using an extension to limit your exposure to hate speech?" was asked but added as a non-required question. This question is aimed at any participants that aren't comfortable with the concept of the HateBlocker extension and gives an opportunity to explain what their reasons are.

5 Results

5.1 Prototype Development

5.1.1 Design

The final extension design was based on the initial design proposed in Section 3.2. Due to limitations caused by the method used to select elements as discussed in Section 3.3.4, a simpler method of replacing the text was used as shown in Figure 26.

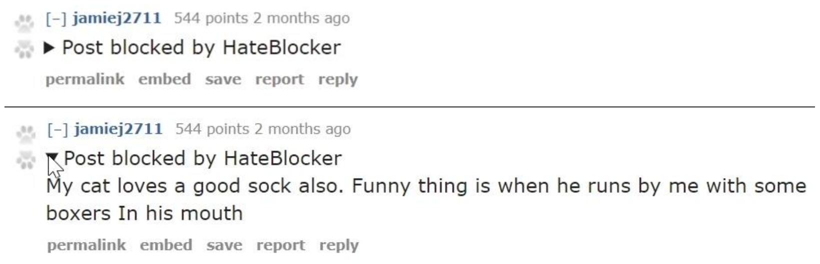


[–] jamiej2711 539 points 1 month ago

My BLOCKED BY HATEBLOCKER loves a good sock also. Funny thing is when he runs by me with some boxers In his mouth

Figure 26 - Initial appearance of blocked elements

After receiving qualitative feedback via the survey described in Section 4.2.4, this decision was discarded for a more minimal notification, after participants voiced that they'd rather not see the notice prominently on the page, instead blending in more with the page background. This method also allowed for users to view the content if they wished by selecting the dropdown as shown in Figure 27.



[–] jamiej2711 544 points 2 months ago

▶ Post blocked by HateBlocker

permalink embed save report reply

[–] jamiej2711 544 points 2 months ago

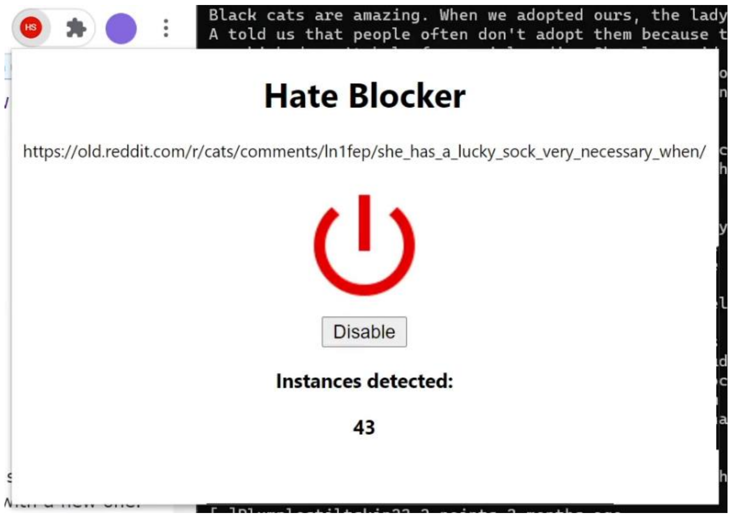
▼ Post blocked by HateBlocker

My cat loves a good sock also. Funny thing is when he runs by me with some boxers In his mouth

permalink embed save report reply

Figure 27 - Final appearance of blocked elements

The extension's pop-up interface as shown in Figure 28 was also based on the original design.



https://old.reddit.com/r/cats/comments/ln1fep/she_has_a_lucky_sock_very_necessary_when/

Instances detected:

43

Figure 28 - Final appearance of pop-up

5.1.2 User Experience

As demonstrated in the Final Demonstration artefact, the extension is used in tandem with a Python server. This server could be configured remotely but for the proof of concept it was ran locally alongside the extension. When the extension is installed the logo can be clicked to open up the pop-up interface, and this is where the extension's blocking feature is activated.

The proof of concept version is designed to run on the website "old.reddit.com", and when the page is visited, elements that contain words that are present in the blocked_words CSV file, or go over the toxicity threshold of Perspective API, are blocked.

If the extension's pop-up is selected while the blocking feature is active, the number of blocked elements on the page are displayed. If a link is clicked to a different page, this process repeats with the number of blocked elements being refreshed for the updated page. The extension can also be disabled from this pop-up interface.

5.1.3 Performance of models

The local word detection method used converts all characters to lowercase, so has a 100% theoretical accuracy rate when running from a CSV file.

As the Perspective API responded with a toxicity score between 0 and 1, a threshold of 0.5 was used to determine whether the element is blocked or not. This value was decided on as a sensible middle ground that didn’t block harmless comments, whilst not letting more harsh toxic comments through.

5.2 Survey

In total, 124 participants from a wide range of demographics completed the Google Forms research survey by filling out their demographic information, viewing the demonstration video and filling out the System Usability Scale and providing textual feedback. This section presents the results found, including demographics and data.

5.2.1 Demographics

Of the 124 participants that completed the survey, just over 50% were aged 18-24, with all age groups presented in Figure 29.

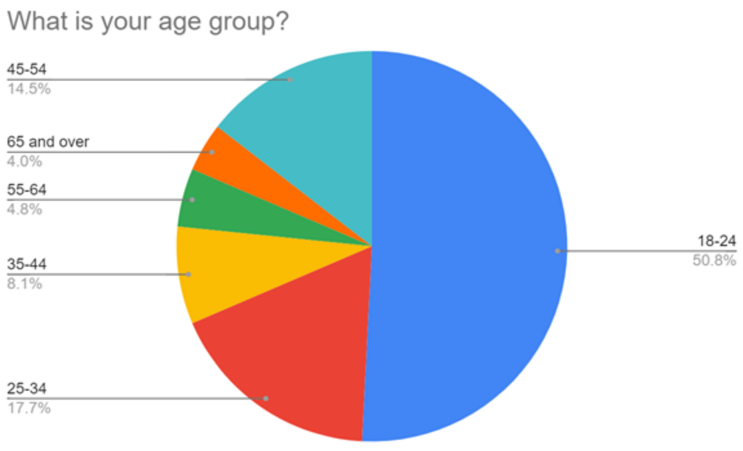


Figure 29 - Age groups of survey participants

The ratio of male to female was relatively even, with these genders making up the majority, and 11% identifying as non-binary. Two respondents answered with specific answers, so in order to protect anonymity these were grouped under “Other”.

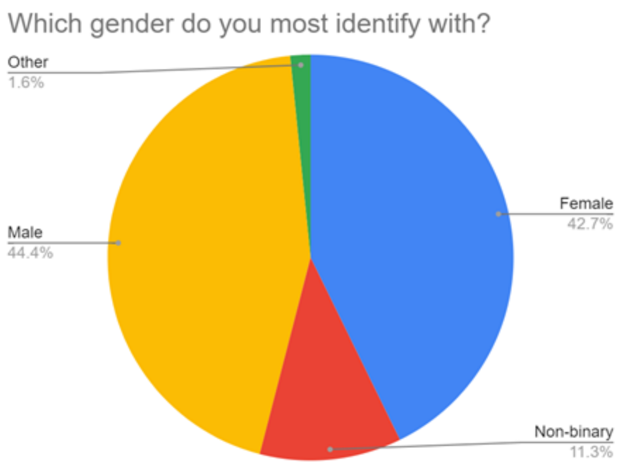


Figure 30 - Identified gender of survey participants

The highest level of education of the majority of the participants is University at 50%, followed by Highers/further education at 30% and Post-graduate degree at 9.7% as shown in Figure 31.

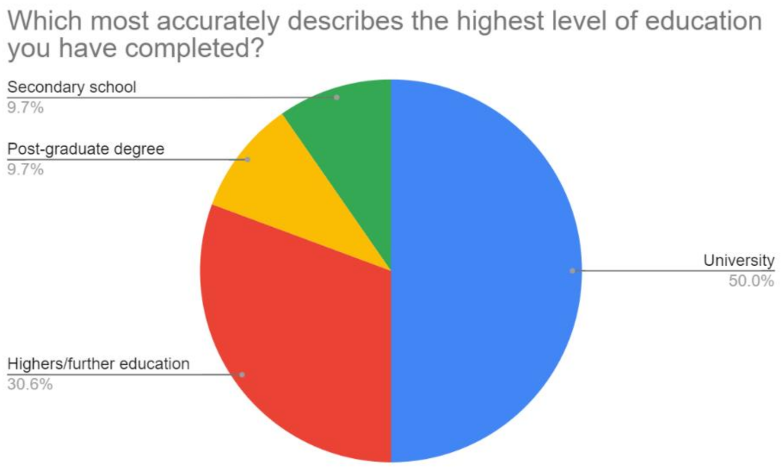


Figure 31 - Level of education of survey participants

Although not strictly demographic information, participants were asked which social media platforms they use, to determine which platforms would be most useful to extend functionality to in the future. Users were encouraged to tick multiple boxes for this section of the survey as shown in Figure 32.

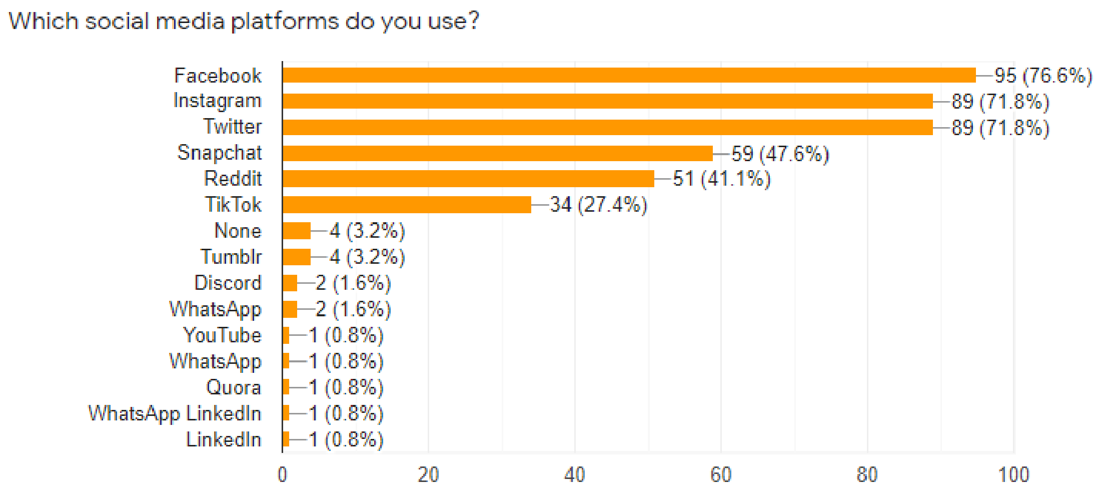


Figure 32 - Social media platforms used by survey participants

5.2.2 System Usability Scale

The 124 responses were analysed to see which areas of the System Usability Scale stood out as strengths and weaknesses. The final score was found to be 77.3 out of 100, resulting in an above average score. As mentioned in Section 4.2.3, research was referenced to find the percentile of this System Usability Scale result.

Figure 33 shown below displays the total responses for each question, keeping in mind that odd numbered questions are phrased positively meaning Strongly Agree is the best answer, with even numbered questions phrased negatively and therefore Strongly Disagree being the desirable result.

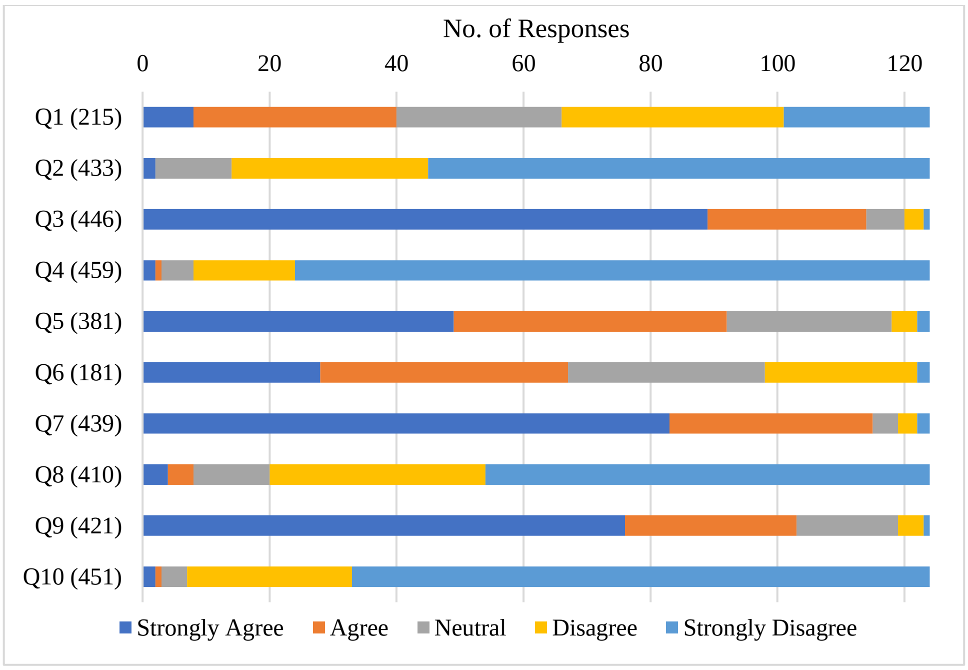


Figure 33 - Stacked bar graph showing System Usability Scale responses

Normalized scores for the questions are shown on the vertical axis, which takes account of the different phrasing and scores the questions on how positive the results were. The following questions scored the highest cumulatively:

Q4 - "I think I'd need the support of a technical person to be able to use this extension."

Q10 - "I would need to learn a lot of things before I could get going with this extension."

The following questions cumulatively scored the worst:

Q6 - "The extension was designed in an aesthetically pleasing way."

Q1 - "I think I would use this extension frequently."

6 Discussion

This section will discuss the project as a whole, relating back to the aim and research questions set out in the Introduction. This discussion will involve the design and development of the browser extension, server, and analysis of the survey methodology and results received.

6.1 Design

A high priority when designing the browser extension was ensuring it was accessible to users of all levels of technical proficiency. This was done by making the purpose of the extension clear through making the interface as simple as possible as well as providing materials to assist users in understanding how the extension works via a demonstration video.

The extension was developed through multiple iterations, with the first version demonstrated to survey participants in a Demonstration Video covered in Section 4.2.2. The majority of the feedback regarding this version involved the fact that the text was too intrusive, as well as users being able to "fill in the blanks" of the blocked word since the rest of the sentence was visible, with over 30 responses mentioning this:

"Perhaps find a way to block out sentences after the blocked word as you will know the word you have blocked out but an Excellent first step."

"The capital letters reading BLOCKED BY HATEBLOCKER along with the yellow highlight were a bit obtuse to the eye and draw attention to themselves a little too much"

This was addressed with the final version of the extension, where as shown in Figure 27, the highlighting was removed, instead implementing a drop-down that hides the entire sentence, with a button to expand the comment if desired.

In the initial design of the extension, it was proposed to show the user the specific type of hate speech the extension was hiding, but due to the method of word detection used, the extension could only tell whether an element was hateful or not, and not the specific reason it was. Therefore, the number of instances on the page were displayed to inform the user. This functionality could be implemented in the future by using a local ML model as discussed in Section 7.1.

Survey participants praised the overall ease of use of the extension, with Q2, 3, and 4 related to the level of knowledge required to use it, with Q4 - "I think I'd need the support of a technical person to be able to use this extension" - the highest scoring answer with an average response of 1.2 and the majority of participants choosing Strongly Disagree which is a sign that users understand the purpose and usage of the extension, leaving responses such as:

"It looks clean and easy to use as is."

6.2 Technical Implementation

A core objective of this project was to develop a browser extension that would be able to detect and block hateful content on webpages. The extension was successful in this regard in that it effectively detected relevant elements of the target page, sending these using an API to communicate with the server that processed these results. It was also used to inform users of the status of the extension, and the number of instances of hate speech on any particular page.

A common point of feedback found in survey results regarding the technical functionality of the extension had to do with integrating a "word-list" or "machine learning" that could be used instead of having to manually enter a single hateful word to block:

"Could potentially implement a list of pre banned words, similar to the kind of thing you get when creating usernames on most accounts."

"I think the proposed method of adding words to a block list or using machine learning would be a good method to block words however I would like to see a user manageable list as some people have different qualifiers as what is hate to them."

This functionality – Specifically a word-list and cloud based sentiment analysis powered by the Perspective API, was implemented in the final version of the extension, as explained in Section 3.3.3

A targeted approach was used when building the detection functionality of this extension, and this significantly simplified the process of detecting elements on the chosen site, where the Paragraph tag was used. A downside of this method of element detection was that a lot of irrelevant elements were captured and processed such as page headings and navigation/side bars. This didn't cause issues when carrying out local word detection, as the API traffic stayed on the local machine, however if this server was moved to a remote location, it's possible the network connection would be a bottleneck.

Along with this extension, a server was implemented to carry out the processing function of the browser extension, enabling the two systems to exist independently. This would allow the browser extension to be modified and configured with different detection models, and the server to be used with any application that wishes to integrate with the HTTP API.

Python was the language of choice for this server, due both to the simplicity of configuring an API using Flask and the wide range of libraries available for text processing and Machine Learning. The extension was built with an intent to make the process of integrating new detection algorithms simple, with a standard JSON input format of `sentence:true/false` and each method contained within a function inside the analyser.

The server-side approach was successful in that it integrated well with the extension and performed well when working with a local word detection model.

Problems were experienced however when working with the remote Perspective API, where due to every paragraph element of the page being individually processed, often the rate limit of 60 requests per minute was reached before classifying all the elements on the page. Although the same rate limits were not experienced with the similar Google Cloud Language, this API was less catered towards toxicity and similarly to Perspective API, the time spent waiting for a response made these methods unusably slow on websites that were more complex.

6.3 Survey

The survey was published successfully with more members of public taking part than expected, with the researcher aiming for 50 responses and receiving 124. Although a relatively diverse range of participants were reached, bias was found in that the participants were disproportionately highly educated with 59% of participants having a university degree or higher, with 32% of the Scottish population having achieved the same as of 2018 (Statista, 2019). As well as this, 50% of the participants that responded were 18-24. This bias can be explained by the fact the survey was shared via personal social media pages where followers were disproportionately young and educated.

The method of an online survey with a demonstration video was found to be effective for receiving a large amount of responses in a short amount of time, with the Google Form receiving 124 responses within 5 days of publishing. Receiving feedback quickly allowed for modifications to be made to the extension, after analysing feedback and finding unanimous sentiment regarding highlighting of hateful words for example.

Although making up 42% of total survey responses, Female identifying participants were much more likely to be uncomfortable with their exposure to hateful content online, as shown in Figure 34. This is supported by literature reviewed, further proving the unequal consequences of hateful speech online.

Q11 - "I'm comfortable with my amount of exposure to hateful content online."

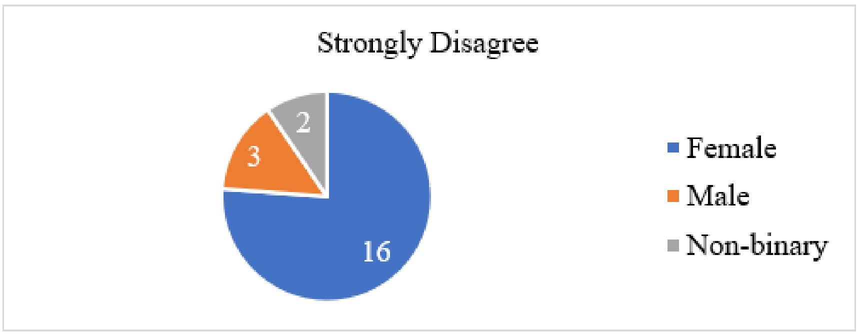


Figure 34 – "Strongly Disagree" Responses to survey Q11

Overall, as shown in Figure 25, the System Usability Score of 77 would fall under the category of "good". This is a positive finding and proves the extension has potential as part of a wider approach of reducing hate speech online.

Regarding the popularity of different platforms, Facebook, Instagram, and Twitter were found to be the most popular amongst participants, which was expected as Facebook and Instagram have over 1 billion monthly users. Conversely however, when asked "Which websites would you see this extension being useful on?", Twitter was mentioned 48 times with Facebook in 2nd place with 38 mentions, suggesting users encounter hateful content on Twitter more often versus other platforms.

Q1 of the survey related to how often participants believed they'd use the extension, with a mean average result of 2.75 and "Disagree" being the most popular response. Participants that were uncomfortable with their level of exposure to hate speech scored this question with an average score of 3.2, which could suggest that users who are exposed to this content more often are more likely to use the extension.

7 Conclusion

The primary aim of this project involved the investigation of how a browser extension can be used to reduce exposure to hate speech online. It found, through investigation of existing literature, that NLP can play a valuable role in hate speech detection due to its ability to recognise context in speech, and how browser extensions can be an effective method of managing what users are exposed to online. Through the development and evaluation of a proof of concept extension, it was found that users were receptive to this method of reducing their hate speech exposure online.

The extension was designed to be simple for the end user, whilst maintaining a level of interoperability and extensibility to add functionality in the future. It was found that overall, the extension was highly usable with a System Usability Score of 77, and a standard API allowed for the server or extension to be modified without having to make major changes to the protocol.

As well as this, survey results suggest that the public are open to using browser extensions or similar solutions to reduce their exposure to hate speech online, with the majority of negative feedback related to the fact that the demonstration was based on an early proof of concept version of the extension.

It was found that sentiment analysis and NLP in general could massively assist in reducing the amount of hateful speech online, especially when online platforms integrate it to deal with toxic accounts directly. Browser extensions, however, have the potential to play an important role for individuals that wish to cater their online experience separate from what online platforms deem as acceptable.

Ultimately the research questions were fulfilled with literature and technologies reviewed, in order to gain required knowledge used when developing a prototype extension. Finally, the general public were surveyed on a demonstration of this solution.

7.1 Limitations and Future Work

In order to manage the practical workload of this project, some technical solutions that could have been integrated into the project weren't implemented. This section will detail some potential solutions and how this project could be expanded in the future.

A limitation present in the current version of the HateBlocker project was the choice made between having a less accurate and context unaware word detection model that processes quickly, or a more accurate context aware sentiment analysis model that hinders performance. The decision was made not to include a local ML model for the proof of concept due to the Perspective API being perceived as more accurate and being easier to implement, but after testing was found to be unusably slow when used with complex webpages. Sentiment analysis could be integrated to achieve a similar level of responsiveness as word-detection if implemented locally, while also opening the possibility of giving context-dependant warnings proposed in the initial design shown in Section 3.2.

As well as this, the extension's usefulness is restricted by the limitation of being usable only on "old.reddit.com", a choice made due to the static nature of the old site making it easier to develop on top of. As shown in survey results, participants most often answered Twitter as the platform this extension would be useful on. Although the current selection method based on a static DOM would be insufficient for this site, due to the modular nature of the project and the existing Shinigami Eyes extension mentioned in Section 2.3 functional on Twitter, it would likely be possible with further research.

More widely, the extension could be expanded to other browsers such as Mozilla Firefox and Microsoft Edge. A larger challenge would be bringing this technology to increasingly popular mobile platforms where content injection is impossible in most instances, with the exception of Firefox for Android offering limited extension functionality.

References

Affairs, A. S. for P. (2013) *System Usability Scale (SUS)*. Department of Health and Human Services. Available at: system-usability-scale.html (Accessed: 26 April 2021).

Bangor, A. (2009) 'Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale', 4(3), p. 10.

Barendt, E. (2019) *What Is the Harm of Hate Speech? | SpringerLink*. Available at: https://link.springer.com/article/10.1007/s10677-019-10002-0 (Accessed: 28 April 2021).

BBC News (2019) 'Zuckerberg outlines plan for "privacy-focused" Facebook', 7 March. Available at: https://www.bbc.com/news/world-us-canada-47477677 (Accessed: 5 May 2021).

Carnegie Mellon University (no date) *Useful Resources*. Available at: https://www.cs.cmu.edu/~biglou/resources/ (Accessed: 29 April 2021).

Diagrams.net (2020) *Diagram Software and Flowchart Maker*. Available at: https://www.diagrams.net/ (Accessed: 29 April 2021).

Edmonds, W. A. and Kennedy, T. D. (2017) *An Applied Guide to Research Designs: Quantitative, Qualitative, and Mixed Methods*. 2455 Teller Road, Thousand Oaks California 91320: SAGE Publications, Inc. doi: 10.4135/9781071802779.

EIU (2020) *Measuring the prevalence of online violence against women*, *Jigsaw Infographic*. Available at: https://onlineviolencewomen.eiu.com/ (Accessed: 4 May 2021).

Facebook (2021a) *Community Standards | Facebook*. Available at: https://en-gb.facebook.com/communitystandards/hate_speech (Accessed: 4 May 2021).

Facebook (2021b) *React – A JavaScript library for building user interfaces*. Available at: https://reactjs.org/ (Accessed: 25 April 2021).

Fišer, D. *et al.* (eds) (2018) *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium: Association for Computational Linguistics. Available at: https://www.aclweb.org/anthology/W18-5100 (Accessed: 30 March 2021).

Flask (2020) *Welcome to Flask — Flask Documentation (1.1.x)*. Available at: https://flask.palletsprojects.com/en/1.1.x/ (Accessed: 26 April 2021).

Gogarty, K. (2020) *A new study finds that Facebook is not censoring conservatives despite their repeated attacks*, *Media Matters for America*. Available at: https://www.mediamatters.org/facebook/new-study-finds-facebook-not-censoring-conservatives-despite-their-repeated-attacks (Accessed: 3 May 2021).

Google (2014) *Google Forms: Free Online Surveys for Personal Use*. Available at: https://www.google.co.uk/forms/about/ (Accessed: 22 April 2021).

Google (2019) *Toxicity*, *Jigsaw*. Available at: https://jigsaw.google.com/the-current/toxicity/ (Accessed: 4 May 2021).

Hall, R. H. and Hanna, P. (2004) 'The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention', *Behaviour & Information Technology*, 23(3), pp. 183–195. doi: 10.1080/01449290410001669932.

Hatebase (2021). Available at: https://hatebase.org/ (Accessed: 29 April 2021).

Howard, J. W. (2019) 'Free Speech and Hate Speech', *Annual Review of Political Science,* 22(1), pp. 93–109. doi: 10.1146/annurev-polisci-051517-012343.

Inflectra (2021) *What is Waterfall & Hybrid Development? | Methodologies | Inflectra*. Available at: https://www.inflectra.com/methodologies/waterfall.aspx (Accessed: 1 May 2021).

Jack Bowker (2021) *Hate Speech Blocker Browser Extension Demo*. Available at: https://www.youtube.com/watch?v=6Z8McJWQFLc (Accessed: 23 April 2021).

Jain, S. and Kamthania, D. (2020) *Hate Speech Detector: Negator*. SSRN Scholarly Paper ID 3563563. Rochester, NY: Social Science Research Network. doi: 10.2139/ssrn.3563563.

Kapil, P., Ekbal, A. and Das, D. (2020) 'Investigating Deep Learning Approaches for Hate Speech Detection in Social Media', *arXiv:2005.14690 [cs]*. Available at: http://arxiv.org/abs/2005.14690 (Accessed: 30 March 2021).

Kumar, R. *et al.* (2018) ‘Benchmarking Aggression Identification in Social Media’, in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1–11. Available at: https://www.aclweb.org/anthology/W18-4401 (Accessed: 30 March 2021).

Kuniecki, M., Pilarczyk, J. and Wichary, S. (2015) ‘The color red attracts attention in an emotional context. An ERP study’, *Frontiers in Human Neuroscience*, 9. doi: 10.3389/fnhum.2015.00212.

Lee, Y.-S. (2018) ‘Analysis on Trends of Machine Learning-as-a-Service’, *International Journal of Advanced Culture Technology*, 6(4), pp. 303–308. doi: 10.17703//IJACT2018.6.4.303.

Leets, L. (2002) ‘Experiencing Hate Speech: Perceptions and Responses to Anti-Semitism and Antigay Speech’, *Journal of Social Issues*, 58(2), pp. 341–361. doi: https://doi.org/10.1111/1540-4560.00264.

MacAvaney, S. *et al.* (2019) ‘Hate speech detection: Challenges and solutions’, *PLoS ONE*, 14(8). doi: 10.1371/journal.pone.0221152.

MAGIX (2016) *Video Production Software — Unleash Your Creativity | VEGAS*. Available at: https://www.vegascreativesoftware.com/gb/ (Accessed: 23 April 2021).

Malmasi, S. and Zampieri, M. (2017) ‘Detecting Hate Speech in Social Media’, *arXiv:1712.06427 [cs]*. Available at: http://arxiv.org/abs/1712.06427 (Accessed: 3 March 2021).

Matal v. Tam (2017) *S. Ct.*

Modha, S. *et al.* (2020) ‘Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance’, *Expert Systems with Applications*, 161, p. 113725. doi: 10.1016/j.eswa.2020.113725.

Mubarak, H., Darwish, K. and Magdy, W. (2017) ‘Abusive Language Detection on Arabic Social Media’, in *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 52–56. doi: 10.18653/v1/W17-3008.

Newcomer, E. (2020) *Racism is rampant on Reddit, and its editors are in open revolt, Fortune*. Available at: https://fortune.com/2020/06/20/racism-reddit-editors-in-revolt/ (Accessed: 25 April 2021).

OBS (2021) *Open Broadcaster Software | OBS*. Available at: https://obsproject.com/ (Accessed: 22 April 2021).

Oksanen, A. *et al.* (2014) 'Exposure to Online Hate among Young Social Media Users', in *Sociological Studies of Children and Youth*, pp. 253–273. doi: 10.1108/S1537-466120140000018021.

Perspective Developers (2019) *About the API - FAQs*. Available at: https://support.perspectiveapi.com/s/about-the-api-faqs (Accessed: 1 May 2021).

Ross, B. *et al.* (2016) 'Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis', in. doi: 10.17185/duepublico/42132.

Sauro, J. (2011) *SUStisfied? Little-Known System Usability Scale Facts User Experience Magazine*. Available at: https://uxpamagazine.org/sustified/ (Accessed: 26 April 2021).

Stančin, I. and Jović, A. (2019) 'An overview and comparison of free Python libraries for data mining and big data analysis', in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 977–982. doi: 10.23919/MIPRO.2019.8757088.

StatCounter (2021) *Desktop Browser Market Share Worldwide, StatCounter Global Stats*. Available at: https://gs.statcounter.com/browser-market-share/desktop/worldwide (Accessed: 24 April 2021).

Statista (2019) *Education level by gender Scotland 2018 Statistic, Statista*. Available at: https://www.statista.com/statistics/364994/scotland-men-women-education-level/ (Accessed: 1 May 2021).

The Crown Prosecution Service (2017) *Hate crime*. Available at: https://www.cps.gov.uk/crime-info/hate-crime (Accessed: 3 May 2021).

Appendix A – Demographic Information

What is your age group?

- 18-24
- 25-34
- 35-44
- 45-54
- 55-64
- 65 and over

Which gender do you most identify with?

- Male
- Female
- Non-binary
- Other: ___________

Which most accurately describes the highest level of education you have completed?

- Secondary School up to 16 years
- Highers or further education (NC, HNC, etc.)
- University
- Post-graduate degree

Which social media platforms do you use? (Multiple choice)

- Facebook
- Instagram
- Twitter
- Snapchat
- Reddit
- TikTok
- None

Appendix B – Risk Assessment

No.	Risk	Likelihood	Impact	Risk
R1	Research period runs over allocated time	3	2	6
R2	Development of extension basics goes over allocated time	3	4	12
R3	Finding suitable keyword sets takes longer than expected	1	5	5
R4	Development of keyword detection goes over allocated time	5	2	10
R5	Development of natural language model goes over allocated time	6	2	12
R6	Bugs or errors found during testing process	7	1	7
R7	Stress or personal/family issues	4	6	24
R8	COVID-19 rules tighten limiting ability to travel	7	2	14
R9	Unable to find suitable survey participants	3	6	21

R1, R2, R4, R5 - Due to the project being in an area that's not entirely familiar, it's possible these tasks could run over the allocated time due to topics being more complex than planned or having to revise basics. In order to mitigate this risk, the Gantt chart will be updated regularly to reflect how far I believe I am into each task. More time than is estimated to be needed has been allocated to each task to reduce the likelihood of this.

R3 - Although prior research has been done into keyword lists, it's possible that when at the stage of implementing them they're not suitable or not in a suitable format. This risk is mitigated by the fact that there is a large pool of data online that can be sampled, along with English being the best language for these lists.

R6 - It's likely that bugs or errors are found during the development process as well as the testing process. The risk will be mitigated by making sure to make sure all the individual segments of the project operate correctly during the development process to minimise the major bugs found in testing.

R7 - It's possible due to the work from home environment and busy nature of 4th year that stress will be experienced personally. It's also possible for other issues to affect mood and productivity. This risk will be mitigated by taking regular breaks while working and making sure to keep in contact with friends and family.

R8 - Due to lockdown and travelling over the festive period and new year, it's very possible that restrictions could be tightened, and restricted travel could prevent travel between home and Dundee. This risk can't be completely mitigated aside from the fact that the work will be carried out on a laptop which will be on hand at all times.

R9 - It's important for the evaluation of this project that suitable survey participants that are familiar with online environments are found. The risk of not finding suitable participants will be mitigated by planning in advance and allowing for participants to install the extension and take the survey remotely without needing to make face to face contact.

Appendix C – GDPR Sign Off Form

For undergraduate or postgraduate student projects supervised by an Abertay staff member.

This form MUST be included in the student's thesis/dissertation. Note that failure to do this will mean that the student's project cannot be assessed/examined.

Part 1: Supervisors to Complete

By signing this form, you are confirming that you have checked and verified your student's data according to the criteria stated below (e.g., raw data, completed questionnaires, superlab/Eprime output, transcriptions etc.)

Student Name:	Jack Bowker		
Student Number:	1803838		
Lead Supervisor Name:	Dr Jacques Ophoff		
Lead Supervisor Signature	J OPHOFF		
Project title:	Development of a Tool to Reduce Exposure to Hateful Content Online		
Study route:	PhD ☐	MbR ☐	MPhil ☐
	Undergraduate ☑	PhD by Publication ☐	

Part 2: Student to Complete

	Initial here to confirm 'Yes'
I confirm that I have handed over all manual records from my research project (e.g., consent forms, transcripts) to my supervisor for archiving/storage	JB
I confirm that I have handed over all digital records from my research project (e.g., recordings, data files) to my supervisor for archiving/storage	JB
I confirm that I no longer hold any digital records from my research project on any device other than the university network and the only data that I may retain is a copy of an anonymised data file(s) from my research	JB
I understand that, for undergraduate projects, my supervisor may delete manual/digital records of data if there is no foreseeable use for that data (with the exception of consent forms, which should be retained for 10 years)	JB

Student signature : **Jack Bowker**

Date: **07/05/2021**