Hate Speech Detection Extension Model

Ashutosh Mishra, Abhimanyu Singh and Harsh Vyas

Galgotias University, Greater Noida, UttarPradesh, India

Abstract:

In this contemporary world, diversity is far spread. In this background tolerance and moderate ideology are the only tools for peaceful survival. However, it is unfortunate to experience hate speeches quite often. In this background, an attempt has been made to develop a project to detect hate speech on the internet. The project has been developed using HTML, CSS, JavaScript and .Json.

Keywords: Natural Language Processing (NLP), Web Extension.

1. Introduction

A web browser functions as a tool for accessing information on the World Wide Web and provides additional features like bookmarks, screenshots, and history tracking (Boswell, 2017). Companies such as Google, Mozilla, and Microsoft have created their own browsers, available with operating systems or as downloads. Users may not need all services simultaneously, so browser extensions, small add-ons, enhance browser functionality (Hoffman, 2017). These extensions offer third-party functionalities customizable to user preferences (Varshney et al., 2018). Web extensions are software installable on a web browser, represented by an icon on the toolbar. They operate automatically based on browsing activity or by clicking the extension icon, depending on the functionalities built into the extension software.

Browser development companies maintain extension stores for users to download and add extensions to their browsers. Popular extensions worldwide include Grammar, Evernote, ad blockers, and YouTube Downloader (Corpuz, 2017). Han et al. (2010) introduced dins Editor, a browser-based authoring tool as a browser extension, enabling the authoring environment to utilize web page resources efficiently. Marouf et al. (2012) implemented a runtime framework monitoring and controlling permissions by third-party Chrome extensions. Correa et al. (2013) presented the Samekana extension for the Google Chromium Web Browser, facilitating annotation and saving of web references for easy citation in issue tracking system comments.

Phawade et al. (2016) created the "ClickProtect" browser extension for secure browsing, addressing multiple Clickjacking attacks. Khalid et al. (2017) implemented the Google Image Suggestions (GIS) extension to enhance user understanding of terms in various domains. Hao et al. (2018) introduced GSCleaner, a Google Chrome extension for discovering miscategorized entities. Sivanesan et al. (2018) developed a browser extension for Google Chromium to track various attack vectors, alerting users to potential XSS attacks. Kabir et al. (2019) created a Google Chrome extension to verify online texts, highlighting text in different colors based on specified categories, useful for authenticating laws, constitutions, and government documents. This paper introduces an antidote for internet hate speech.

2. Methodology

Users will download and install the web extension from their browser's extension store (e.g., Chrome Web Store, Firefox Add-ons).
Upon installation, users can activate the extension via a toggle switch. The extension may also provide customization options, allowing users to set preferences such as sensitivity levels for hate speech detection.
As users navigate web pages, the extension actively monitors and scans the text content on those pages. The extension employs a hate speech detection model to analyze the text content for potentially offensive language or hate speech. This model may include a combination of rule-based systems and machine learning classifiers trained on a diverse dataset. When the extension detects hate speech, it triggers a response to identify and flag the offensive content. This step ensures that the extension accurately recognizes instances of hate speech.
In cases where hate speech is identified, the extension replaces the offensive words or phrases with asterisks (*) or another predetermined symbol. This step is crucial for preventing the display of harmful content.
The extension provides a visual notification or indicator to the user, signaling that hate speech has been detected and modified. This transparency keeps users informed about the extension's actions. Optionally, the extension may log instances of detected hate speech for the user's reference or provide a reporting mechanism. This information can be valuable for users seeking to understand the prevalence of hate speech on specific websites. Users have the option to interact with the extension, accessing a settings panel to modify preferences, view logs, or report false positives/negatives. This step ensures user engagement and allows for fine-tuning of the extension based on user feedback. The extension continuously monitors web pages in real-time as users browse the internet, providing ongoing protection against hate speech. Periodically, the extension receives updates to its hate speech detection model. These updates may include improvements based on emerging linguistic patterns or changes in hate speech trends.
The extension incorporates ethical considerations, such as user consent, transparency in operation, and mechanisms to prevent misuse. The customization options empower users to align the extension with their preferences

2.1 Category of hate speech

To find out the category for a speech which may or may not be a hate speech. First, we find out the intensity of hate speech and if it's greater than 60% which means it's a hate speech, using the intensity we plot it on a graph where we apply classification to find out the closest neighbour of the already classified categories. Once we have found the closest neighbours, we can then use sentiment attached to each category to find out the actual category of the hate speech. Some hate speech categories provided by negator's hate speech model are: - Abuse, Personal Attacks, Cyberbullying, Sexual Advances, Bigotry, Criminal Activity, Death Threats.

3. Working

- Our extension is installed and detecting all hate speech words in the webpage in this case google.
- All the read data is wrapped up in special HTML tags which mark the start and ending of the (with ID) and of the post separated by a new line.
- Now, we can add any custom word in the text box given to block any offensive word (even if it is in a native language).
- The word will be replaced by asterisk (*) symbol in all over the webpage.

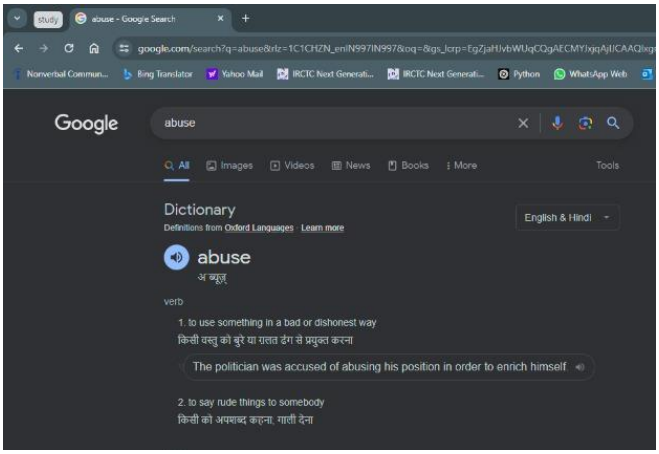

Fig. 1 – Content with no filtering

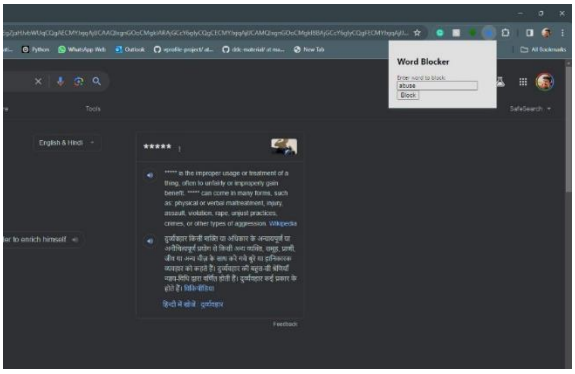

Fig. 2 - Extension UI

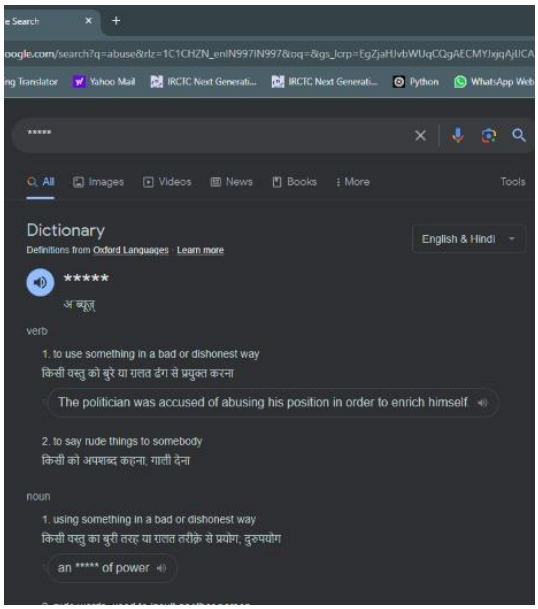

Fig. 3 - Filtered Content

4. Conclusions

In conclusion, the hate speech detection web extension project represents a proactive and user-centric approach to fostering a safer online environment. By leveraging advanced technologies, including a carefully trained hate speech detection model, the extension works seamlessly in the background, actively monitoring and analyzing text content on web pages. The integration of customization options, such as sensitivity levels and user preferences, empowers individuals to tailor the extension to their specific needs, striking a balance between accuracy and user control.

The replacement of identified hate speech with asterisks not only mitigates the immediate impact of offensive language but also provides users with a clear indication of the extension's actions through visual notifications. The optional logging and reporting features contribute to user engagement and allow for continuous refinement of the extension's performance.

Ethical considerations play a pivotal role in the project, with a focus on transparency, user consent, and prevention of misuse. The extension's interface and documentation aim to educate users about the purpose, limitations, and responsible use of hate speech detection technology.

The project's iterative development process, involving testing, user feedback, and model updates, ensures adaptability to evolving linguistic patterns and emerging challenges in online communication. By deploying this hate speech detection web extension, users gain a valuable tool in promoting a more inclusive and respectful online experience.

In essence, this project aligns with the broader societal goal of addressing online hate speech, offering a tangible solution that combines technology, user empowerment, and ethical considerations. As the digital landscape evolves, the hate speech detection web extension stands as a testament to our commitment to creating a virtual space where individuals can engage in discourse free from the harmful impacts of hate speech.

REFERENCES

W. Boswell. (2017). What is a web browser? Available: https://www.lifewire.com/what-iswebbrowser-3483197

C. Hoffman. (2017). Beginner geek: Everything you need to know about browser extensions. Available: https://www.howtogeek.com/169080/beginner-geekeverything-you-need- to-knowabout-browserextensions/

G Varshney, S Bagade and S Sinha (2018). Malicious browser extensions: A growing threat: A case study on Google Chrome: Ongoing work in progress, International Conference on Information Networking (ICOIN), 10-12 Jan. 2018, Thailand. DOI: 10.1109/ICOIN.2018.8343108

J. Corpuz. (2017). 41 Best Google Chrome Extensions. Available: https://www.tomsguide.com/us/picturesstory/283-best-google-chrome-extensions.html

Han, S., Kim, J., Lee, Y., Cha, J., & Choi, B.-U. (2010). dinsEditor: A Browser Extension for QTICompliant Assessment Item Authoring. 2010 10th IEEE International Conference on Advanced Learning Technologies. doi:10.1109/icalt.2010.71

Marouf, S. M., Shehab, M., & Desikan, A. (2012). REM: A runtime browser extension manager with fine-grained access control. 2012 Tenth Annual International Conference on Privacy, Security and Trust. doi:10.1109/pst.2012.6297947

Correa, D., Lal, S., Saini, A., & Sureka, A. (2013). Samekana: A Browser Extension for Including Relevant Web Links in Issue Tracking System Discussion Forum. 2013 20th Asia-Pacific Software Engineering Conference (APSEC). doi:10.1109/apsec.2013.15

Shardul Katare, Saurabh Bawdhankar, Sourab Patil, Aniket Deshpande and Shrikat Nagure (2016), Google Chrome Extension for Topic Summarization, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 1, pp. 422-425, January 2016, ISSN: 2277 128X

Pawade, D., Reja, D., Lahigude, A., & Johri, E. (2016). Implementation of extension for browser to detect vulnerable elements on web pages and avoid Clickjacking. 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence). doi:10.1109/confluence.2016.7508118

Hao, S., Xu, Y., Tang, N., Li, G., & Feng, J. (2018). Cleaning Your Wrong Google Scholar Entries. 2018 IEEE 34th International Conference on Data Engineering (ICDE). doi:10.1109/icde.2018.00185

Sivanesan, A. P., Mathur, A., & Javaid, A. Y. (2018). A Google Chromium Browser Extension for Detecting XSS Attack in HTML5 Based Websites. 2018 IEEE International Conference on Electro/Information Technology (EIT). doi:10.1109/eit.2018.8500284