BT5153 Group Project G9 - Measuring Hate Speech for Social Media Platform

Miao Zijun [1] **Wang Chang** [1] **Wang Yexin** [1] **Wang Yili** [1] **Yang Mingke** [1]

Abstract

Abundant online comments on social media platforms have caused the presence of hate speech which may lead to social conflicts, emotional harm, and even long-term negative impacts on individuals and communities. This project aims to provide a tool to detect hate speech across five common categories, including Status, Violence, Genocide, Dehumanization, and Humiliation. We collected 39,565 comments, conducted exploratory data analysis and constructed traditional Support Vector Machine (SVM) model and DistilBERT model. Our findings indicate that the DistilBERT model outperforms traditional SVM model combined with TF-IDF method. Additionally, we utilized LIME to interpret the DistilBERT model and gain insights to common words associated with hate speech. Furthermore, we developed a user-friendly web-based interface for visualization of model predictions and real-word application.

1. Introduction

1.1. Project Overview

This project addresses the critical issue of hate speech prevalent across online comments on various social media platforms. Characterized by its derogatory, inflammatory, or discriminatory nature, hate speech significantly undermines the efforts to foster inclusive and respectful online communities. To combat the pernicious effects of hate speech, this study proposes the development of an automated detection and filtering tool utilizing advanced deep learning techniques.

The core of this solution is the application of neural network architectures, notably DistilBERT, which are adept at processing natural language. These models excel in identifying subtle nuances and contextual cues that are indicative of hate speech, thereby enhancing the detection's accuracy and efficiency.

Our methodology categorizes hate speech into five specific dimensions: HumiliateStatusDehumanizeViolenceand Genocide. These dimensions are selected based on their pronounced impact on societal harm and their prevalence in online platforms. Each category encapsulates a range of harmful content that can incite distress, perpetuate biases, and escalate conflicts (Sachdeva et al., 2022).

For model training, we employed a binary classification approach where labels are assigned as “0” for the absence and “1” for the presence of hate speech attributes. This binary labeling strategy enhances the precision of our model by distinctly categorizing content, facilitating a more effective approach to content moderation. This methodology is pivotal for systematically reducing the prevalence and impact of hate speech on social media platforms, fostering safer and more inclusive digital environments.

1.2. Importance of the Study

The primary objective of this project is to enhance the user experience on social media platforms by effectively identifying and categorizing instances of hate speech. This automated tool will not only aid in maintaining a healthier online discourse but also reduce the operational costs associated with manual content moderation. Additionally, by improving the safety and inclusiveness of online spaces, the tool aims to increase user engagement and satisfaction, ultimately contributing to the growth and success of social media platforms.

This initiative aligns with broader business goals by addressing key challenges in digital communication and community management. It also positions the platform to better comply with increasing regulatory demands for responsible content moderation. The success of this project will be measured through quantitative metrics such as the accuracy and efficiency of the hate speech detection model, as well as its impact on user retention and platform growth.

[*] Equal contribution [1] Master of Science in Business Analytics, National University of Singapore, Singapore. Correspondence to: Miao Zijun <e1148764@u.nus.edu>, Wang Chang <e1148713@u.nus.edu>, Wang Yexin <e1148733@u.nus.edu>, Wang Yili <e1148634@u.nus.edu>, Yang Mingke <e1148728@u.nus.edu>.

2. Data Description

2.1. Data Collection

The data collection process for our project relies on an extensive and detailed dataset publicly released and described in academic studies(Kennedy et al., 2020; Sachdeva et al., 2022). This dataset comprises a rich compilation of 39,565 comments, which were annotated by 7,912 different annotators resulting in 135,556 combined rows of data. This diverse collection is instrumental in providing a nuanced understanding of hate speech across various contexts and demographics.

2.2. Exploratory Data Analysis(EDA)

Exploratory Data Analysis (EDA) is a foundational step in data science that allows analysts to understand the patterns, relationships, and anomalies within their data. In this project, we conducted EDA on a dataset of social media comments annotated for hate speech to gain insights into the prevalence and characteristics of different types of hate speech.

2.2.1. Frequency Distribution of Sentiments Across Labels

We aggregated the data by "comment id" and "speeches" and calculated the mode or mean for each type of hate speech label within these groups. Then, we examined the frequency distribution of these aggregated sentiments across the different labels: "status", "violence", "genocide", "dehumanize", and "humiliate".This analysis was visualized through a bar plot, highlighting the prevalence of each sentiment across the comments. This visualization helps to understand which types of hate speech are most common and how they are distributed across the dataset.

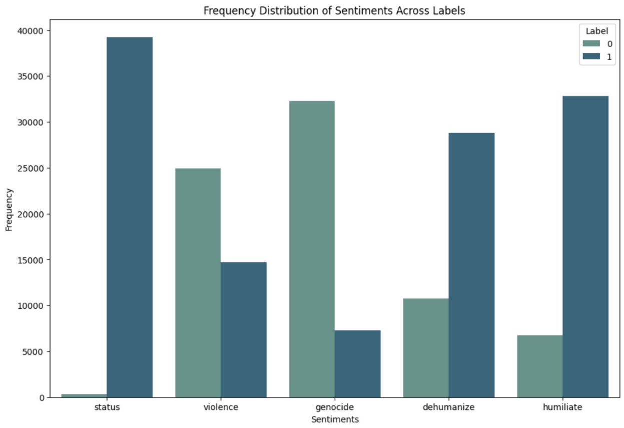


Figure 1. Frequency Distribution of Sentiments Across Five Labels

The bar chart provides a quantitative overview of the frequency of different categories of hate speech across a large dataset of social media comments. The data highlights the prevalence of various forms of hate speech, with 'status' being the most frequently occurring category. This is followed by "humiliation", "violence", "dehumanization", and "genocide", in descending order of frequency. The significant presence of 'humiliation' and 'violence' underscores the severe nature of aggression in online platforms. Although 'genocide' has the lowest frequency among the categories analyzed, its presence is concerning and points to the extreme forms of hate speech that exist within the dataset.

2.2.2. Word Cloud Analysis: The 200 Most Frequent Words in Speeches

To further explore the qualitative aspects of the comments, a word cloud was generated from the 'speeches' column. This visualization illustrates the most frequently mentioned words within the comments, providing insight into the common themes and terms associated with hate speech. This method is particularly useful in identifying prominent words that may need further contextual analysis to understand their usage and implications in hate speech.

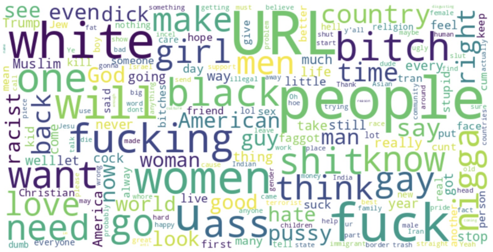


Figure 2. World Cloud for dataset

The word cloud visualization complements the bar chart by providing qualitative insights into the language commonly used in hate speech. Dominant words such as explicit profanities and slurs, particularly those targeting race and sexual orientation, illustrate the aggressive and derogatory nature of the content. The prominence of words related to identity and derogation ("black", "Muslim", "gay") highlights targeted hate speech against specific groups. This visualization effectively captures the harsh and harmful language prevalent in social media discussions, providing a stark representation of the challenges faced in moderating online platforms.

2.2.3. Distribution of speech Lengths

The histogram displays the distribution of lengths of comments (referred to as "speech lengths") captured in the dataset. The x-axis represents the length of comments measured in characters, while the y-axis denotes the frequency of comments corresponding to each length.

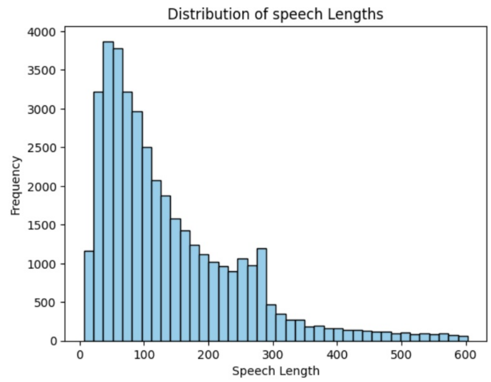


Figure 3. Distribution of Speech Lengths of Dataset

The distribution of speech lengths is predominantly right-skewed, indicating that most comments are relatively short. The peak of the distribution occurs in the 50-100 character range, suggesting that a significant majority of users tend to post brief comments. The frequency steadily decreases as the length of the comments increases, with very few comments extending beyond 300 characters.

3. Solution

3.1. Data Pre-processing

To simplify the classification task, the original labels were transformed into a binary format. Specifically, each label in the columns status, violence, genocide, dehumanize, and humiliate underwent a binary transformation. Labels with a value of 0 were mapped to 0, representing the absence of the corresponding attribute, while any other value was mapped to 1, indicating the presence of the attribute. This binary encoding streamlined the classification problem by converting multi-class labels into a binary classification format, thereby enhancing model interpretability and performance.

Following label transformation, the dataset was aggregated to consolidate redundant information and improve computational efficiency. Utilizing the groupby operation, the dataset was grouped by unique combinations of comment_id and speeches, ensuring that each comment was uniquely represented in the aggregated dataset. Within each group, the values of the label columns were aggregated using a custom function that prioritized the mode value, falling back to the mean value if no mode was available. This aggregation process yielded a refined dataset, named df_grouped, containing unique combinations of comments and their corresponding aggregated label values. By condensing the data while preserving essential information, this aggregated dataset provided a standardized and streamlined foundation for subsequent analysis and model training.

3.2. Machine Learning Models

To address the issue mentioned above, we will compare and evaluate the effectiveness of traditional machine learning models versus modern deep learning models in predicting whether it has hate sentiments based on known reviews and labels, which are the supervised experiments. Initially, we established a baseline using a Support Vector Machine (SVM) model combined with the TF-IDF text vectorization method. Subsequently, we implemented the DistilBERT model, a state-of-the-art transformer-based language model, to determine whether it could enhance performance.

3.2.1. TF-IDF WITH SUPPORT VECTOR MACHINE (SVM)

TF-IDF is a crucial text representation technique in statistical NLP, capturing the significance of words by balancing their frequency in individual documents against their commonness across a corpus. It starts with constructing a Document-Term Matrix, which encapsulates co-occurrence information, and employs the Bag-of-Words model with n-gram features. TF-IDF then refines the importance of each word, with rare terms weighted higher, effectively setting the stage for further analysis.

Support Vector Machine (SVM), a traditional machine learning algorithm, excels in classification tasks, often utilizing the structured vectors produced by TF-IDF. The Document-Term Matrix under TF-IDF transformation provides SVM with rich, differentiated input features, enabling it to identify the optimal separating hyperplane in feature space. Together, the strategic application of TF-IDF with SVM creates a robust framework for predicting ratings from user-generated reviews, as illustrated by the methodical conversion from raw text to insightful features, ready for SVM's discriminative classification, which approaches harnesses the strength of statistical methods in NLP(Luthfi & Lhaksamana, 2020).

3.2.2. DISTILBERT

DistilBERT, a light version of the BERT model, reduces the original 12 transformer layers by half. This modification results in a roughly 40% reduction in size and a 60% increase in speed, while maintaining 97% of BERT's performance across various NLP tasks(Sanh et al., 2020). Consequently, it is particularly well-suited for scenarios where computational resources and time are limited.

When comparing TF-IDF with SVM to DistilBERT, TF-

IDF with SVM stands out for its simplicity in computation and comprehension. It generates numerical representations of text data for input into a well-established classification model. However, it falls short in capturing word semantics and context, struggles with out-of-vocabulary (OOV) words, and grapples with the curse of dimensionality due to the sparse vectors it creates.

In contrast, DistilBERT excels in capturing contextual information and word semantics, and effectively handles OOV words. Nevertheless, it demands significantly more computational resources compared to TF-IDF with SVM.

3.3. Experiment Procedures

3.3.1. TF-IDF WITH SVM

The dataset underwent a systematic splitting process to facilitate model training and evaluation. Initially, the dataset was divided into training and test sets using the train_test_split function from the sklearn.model_selection module, adhering to a 40-30-30 split ratio. The resulting subsets were labeled as df_train df_test, df_validation respectively.

The text data underwent TF-IDF (Term Frequency-Inverse Document Frequency) vectorization using the TfidfVectorizer with English stop words removal and a maximum of 1000 features. This process transformed the textual inputs into numerical feature representations, facilitating subsequent model training.

After applying TF-IDF transformation to the training set, validation set, and test set, we started the model training process. For five different dimensions of comment sentiment, because each dimension is independent of the others, we conducted independent model training.To optimize model performance, we tried several different values of the hyperparameter C, which can illustrate tolerance for sample misclassification and adjust how well the model fits the data. We adopted an SVM model with a linear kernel and performed training and validation set evaluations for each different value of C. During this process, we recorded the score of each model on the validation set.

We can find that when C is greater than 1, the validation score almost never changed, which causes overfitting. So, it is important to retain the best-performing model as the optimal model. In particular, we performed model training and evaluation separately for each classification dimension. Finally, we applied the optimal model for each dimension on the test set to calculate its test score.

3.3.2. DISTILBERT

The dataset splitting process follows the same methodology as described earlier, maintaining a 40-30-30 split ratio for the training, validation, and test sets, respectively. This approach ensures consistency and reproducibility across different phases of model development. Each speech sample is tokenized and encoded using the provided tokenizer from Huggingface's open-source library, which allows for efficient tokenization and encoding of text data.

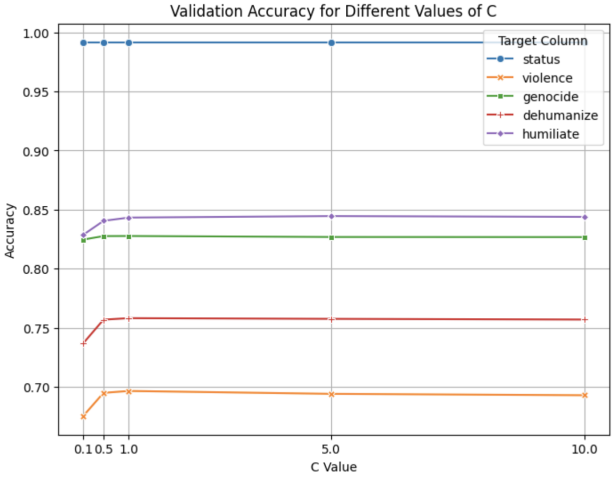


Figure 4. Validation Accuracy for Different Values of C in SVM

The maximum sequence length is set to 512 to align with the input limit of BERT-based models. Our exploratory data analysis (EDA) revealed minimal instances of samples exceeding 512 tokens. Therefore, we truncate samples exceeding this length during subsequent data loader stages, ensuring compatibility with the model architecture.

In contrast to traditional text processing approaches that involve removing stopwords, our methodology with DistilBERT does not require this step. DistilBERT is designed to process complete sentences, enabling it to capture the context and semantics of the text effectively.

The attention mask plays a crucial role in directing the model's attention to relevant tokens while excluding padding tokens. By emphasizing the importance of each token in the input sequence, the attention mask enables the model to focus on meaningful information during both training and inference stages.

By adhering to these preprocessing steps, we prepare the dataset for seamless integration into the DistilBERT model as DataLoader, which optimizing its performance for hate speech classification tasks.

After preparing the DataLoader, we initiated the training process for the pre-trained DistilBERT model from Huggingface's library. We retained the pre-trained layers responsible for feature extraction from text data and finetuned them. Specifically, we augmented the model with an additional fully connected layer at the end, tasked with mapping the distilled features to a binary output. The model's raw output logits from the fully connected layer were passed through a sigmoid function, which converted them into probabilities, allowing us to interpret the model's prediction as the likelihood of a comment being a hate speech (1 indicating presence, 0 indicating absence). This configuration enabled the model to better adapt to the specific binary classification task. Furthermore, we independently initialized the model for each of the five different dimensions of comment sentiment, as each dimension operates independently of the others.

During the training phase, the model undergoes four main steps: forward propagation, loss function calculation, backpropagation, and parameter update. In forward propagation, the model processes the input data and passes it through the pre-trained DistilBERT layers, the additional fully connected layer, and finally through a sigmoid function to obtain the predicted probabilities. Subsequently, the error between these predicted probabilities and the target labels is calculated using the binary cross-entropy loss function, which is suitable for binary classification tasks.

The backward propagation process then computes the gradient of the loss function with respect to each parameter in the model, which is used by the AdamW optimizer to update the parameters. This optimizer, a variant of the Adam optimization algorithm, includes weight decay to help control the learning rate and manage the model's complexity. The entire data set will be trained five times, which indicates that the training epoch equals to five as well. After each training epoch, we evaluate the model on a separate validation set. This is done in evaluation mode, where dropout is not applied, ensuring that the model's performance is assessed based on its full capabilities. No backward propagation or parameter updates occur in this mode, which provides a more accurate assessment of the model's generalization to unseen data.

Upon completion, we plotted the history of training and validation accuracies. The accuracies were plotted, revealing that while training accuracy consistently improved with each epoch, the validation accuracy either plateaued or slightly decreased, suggesting potential overfitting of the model to the training data.

To address this issue, we selected the model with the highest validation accuracy out of the five epochs as the "best model" for making predictions on the test set.

Test scores were calculated in a similar manner to validation accuracy. Throughout the experiment, we utilized a GPU to harness its optimized parallel processing capabilities, which are particularly suitable for the attention mechanism in DistilBERT, thus enhancing the efficiency of our training process.

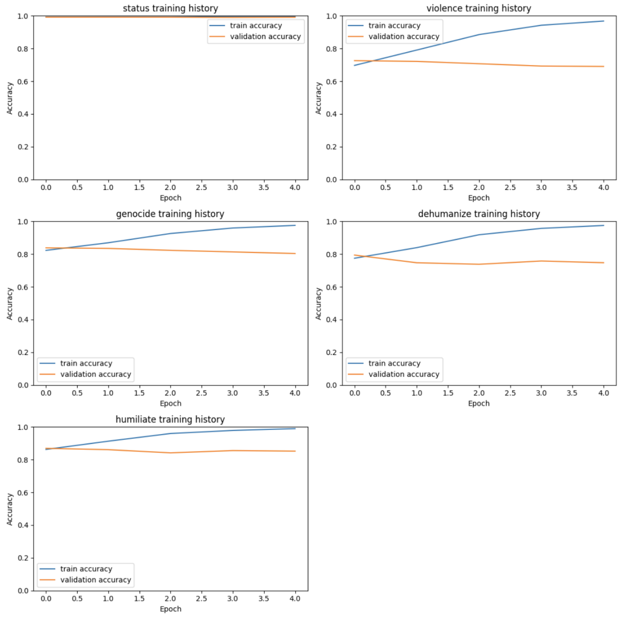


Figure 5. DistilBERT five dimensions Training history

3.4. Results and Comparison

After evaluating both the Support Vector Machine (SVM) and DistilBERT models on the test dataset, we compared their performance across the five evaluation dimensions.

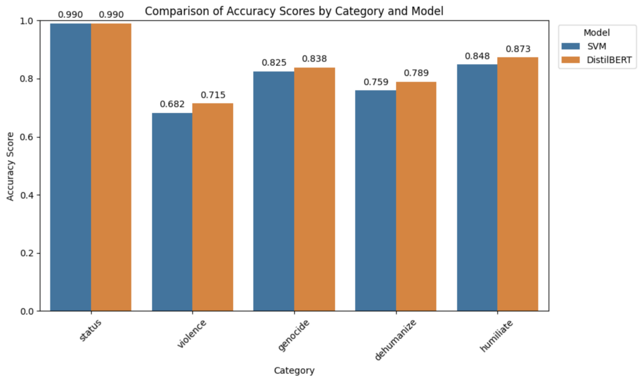


Figure 6. Comparison of Accuracy Scores by Dimensions and Models

The results showed that DistilBERT consistently outperformed the SVM model in terms of accuracy scores across all categories, as depicted in the bar chart. For the "status" category, both models performed similarly well, but Distil-

BERT showed significant improvements over SVM in the "dehumanize", "violence", "genocide", and "humiliate" dimensions.

This improvement suggests that DistilBERT is more adept at capturing nuanced language features present in the training data. Its architecture, based on transformers, allows it to grasp intricate relationships and patterns in text that traditional models like SVM may miss.

Leveraging the superior capability of DistilBERT, we plan to use this model to continue predicting sentiment and analyzing text in future applications. Its ability to understand contextual nuances provides a significant advantage, making it ideal for sentiment analysis tasks where precision and contextual understanding are crucial.

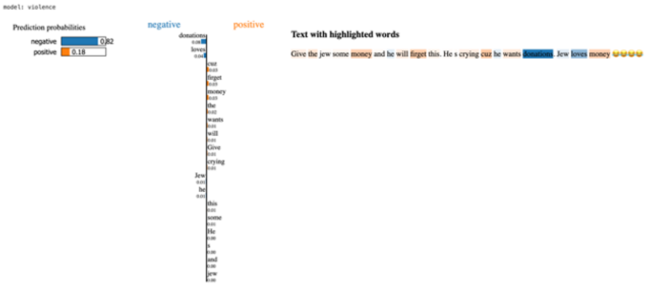
3.5. Explainable Machine Learning

In this section, we explore the interpretability of our models through the application of LIME (Local Interpretable Model-agnostic Explanations). LIME facilitates this process by generating local approximations of complex models through the perturbation of input instances and observing changes in predictions.

Using the LIMETextExplainer, we generated explanations for each model's prediction on the chosen text. Our objective was to spotlight the key features influencing each model's decision-making process. These features serve as indicators of the model's comprehension of the text and offer insights into its internal mechanisms.

To visually illustrate how our model classifies text as positive or negative in different sentiment categories, we randomly select a sample instance from the test dataset for interpretation. Given the sample instance: "Give the jew some money and he will forget this. He's crying cuz he wants donations. Jew loves money". This sentence is labeled as 1 in status, dehumanization, and humiliation while violence and genocide are marked as 0. This implies that the sentence contains hate speech with elements of dehumanization and humiliation. Let's delve deeper into our models to examine different sentiments. As an example, I'll illustrate the results for violence and genocide.

For the violence model, the probability for violence is predicted as 0.18, indicating the absence of violent elements in this sentence, consistent with the true label. The key words influencing the prediction are "donations" and "loves". These words contribute to the negative prediction for violence. This interpretation is highly reasonable because the presence of words like "donations" and "loves" typically suggests a context associated with generosity or affection rather than violence. Therefore, it aligns with the model's prediction of a low probability of violence in the sentence.

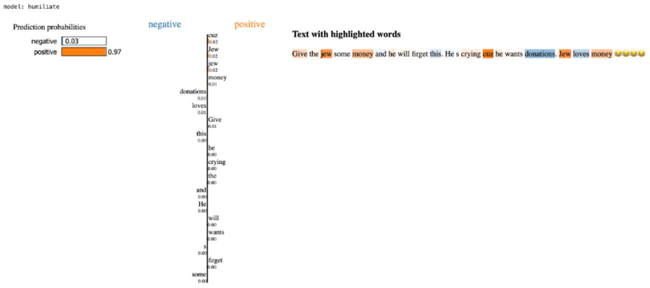

model: violence

Text with highlighted words

Give the jew some money and he will firget this. He s crying cuz he wants donations. Jew loves money 😂😂😂😂

Figure 7. Interface Visualization

Similarly, for the humiliate model, the probability for dehumanization is predicted as 0.97, indicating the presence of elements related to humiliation in this sentence, consistent with the true label. The key words influencing the prediction are "cuz", "jew", and "money." These three words contribute to the positive prediction for humiliation. The words "cuz", "jew", and "money" can be perceived as contributing to a narrative of discrimination or derogation, thus influencing the prediction towards humiliation. Overall, this aligns with our understanding of how certain language cues can indicate elements of dehumanization or humiliation.


model: humiliate

Text with highlighted words

Give the jew some money and he will firget this. He s crying cuz he wants donations. Jew loves money 😂😂😂😂

Figure 8. Interface Visualization

In summary, LIME helps us gain unveiled valuable insights to the effect of each word in one sentence and help us interprete our model better.

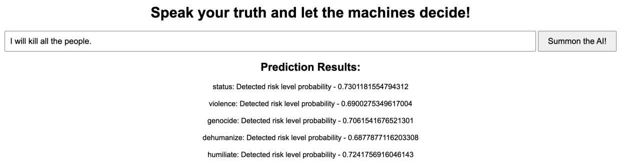
3.6. Interface Visualization

To better utilize our models in real-world scenarios and effectively visualize outcomes, we have developed a web-based interface using Flask and ngrok. Flask, a lightweight Python web framework, is used to create a web server that hosts our predictive interface. Given that we use Colab, a popular cloud-based platform, for training our DistilBERT model using GPUs, there are certain limitations in exposing a local server directly to the internet. Colab doesn't natively support exposing such servers because it runs in a secure, isolated environment.

To address this challenge, we use ngrok, a tunneling service that allows local web servers to be accessible on the internet. By creating a secure tunnel, ngrok provides a public URL that maps to our local server, enabling remote access to our Flask interface. This setup allows us to deploy the interface in Colab and share it publicly via the ngrok URL, ensuring that the model's predictions can be accessed from anywhere with an internet connection.

This approach leverages Colab's computing resources, including GPUs, to train and host models, while ngrok bridges the gap to enable easy sharing of the results through a web interface.

The interface features a simple and intuitive layout with a single input box. Users can enter any text they wish to analyze.



Speak your truth and let the machines decide!

I will kill all the people.

Prediction Results:

status: Detected risk level probability - 0.7301181554794312

violence: Detected risk level probability - 0.6900275349617004

genocide: Detected risk level probability - 0.7061541676521301

dehumanize: Detected risk level probability - 0.6877877116203308

humiliate: Detected risk level probability - 0.7241756916046143

Figure 9. Interface Visualization

This text is then evaluated to estimate the probabilities that it reflects five specific dimensions: status, violence, genocide, dehumanization, and humiliation. Upon entering text and clicking the "Summon the AI!" button, the input is sent to the backend, where it is processed by five trained DistilBERT models, each tailored to one of the specified dimensions and trained to identify relevant text characteristics.

The text is first tokenized using a DistilBERT tokenizer, preparing it for input into the models. Each model processes the input and outputs a raw prediction score. We apply a sigmoid function to these scores to convert them into probabilities, ranging from 0 to 1, where values closer to 1 indicate a higher likelihood of the text pertaining to the respective dimension. These probabilities are then displayed beneath the input box in the web interface, allowing users to easily understand the potential content concerns in the text they have analyzed and providing immediate and clear visualization of the results.

4. Conclusion

In conclusion, this project has developed an effective tool for detecting five common categories of hate speech—Status, Violence, Genocide, Dehumanization, and Humiliation—in comments on social media platforms through the application of advanced data analytics and machine learning techniques. Not only does this tool identify these categories of hate speech, but it also provides insights into their characteristics and contextual nuances.

Throughout this project, we collected online comments, conducted exploratory data analysis, and summarized common themes and terms associated with hate speech through word cloud analysis. This provided valuable insights into online hate speech, allowing for a more informed and targeted approach to hate speech detection. We utilized both traditional SVM model combined with the TF-IDF method and the DistilBERT model to detect the five categories of hate speech in comments. The results demonstrate the superior performance of the DistilBERT model in hate speech detection. Additionally, we incorporated explainable machine learning techniques such as LIME to enhance the interpretability of the DistilBERT model, enabling a deeper understanding of the prediction processes. Furthermore, we developed a user-friendly web-based interface, extending the utility of the tool and facilitating its seamless integration into real-world scenarios.

We hope that this tool will provide social media platforms with an efficient method to detect hate speech in comments, fostering a safe and friendly communication environment. By promptly identifying and removing harmful content, social media platforms can enhance user experience, improve platform safety, mitigate negative impacts, and ensure compliance with relevant regulations. Additionally, by applying machine learning techniques, online community managers can monitor and manage discourse within their communities efficiently, significantly saving manpower and time costs. Furthermore, by utilizing LIME to analyze comments labeled as containing hate speech content, community managers can offer users timely feedback, informing them of inappropriate terms in their comments and promoting a culture of respectful communication.

5. Limitations and Future Work

Before our tool is applied in real business scenarios, there are still areas for improvement.

Firstly, our dataset may not fully represent diverse hate speech in comments across various social media platforms. Accessing a broader range of data sources could enhance the generalizability of our tool. Additionally, extending hate speech detection to multiple languages and cultural contexts can broaden the scope of research and address global challenges. We could explore cross-lingual models and transfer learning techniques to detect hate speech across diverse linguistic landscapes.

Secondly, hate speech evolves over time, and static models may struggle to adapt to emerging trends and linguistic shifts. Developing dynamic monitoring systems capable of continuous learning and adaptation could enhance the effectiveness of hate speech detection in real-time.

Thirdly, while the DistilBERT model showed superior performance compared to traditional SVM model, there may still be room for improvement in terms of accuracy and efficiency. Further fine-tuning model parameters and exploring ensemble techniques could enhance the robustness of our hate speech detection tool.

References

Kennedy, C. J., Bacon, G., Sahn, A., and von Vacano, C. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*, 2020.

Luthfi, M. F. and Lhaksamana, K. M. Implementation of tf-idf method and support vector machine algorithm for job applicants text classification. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 8(2), October 2020. URL https://ejurnal.stmik-budidarma.ac.id/index.php/mib/article/view/2276.

Sachdeva, P., Barreto, R., Bacon, G., Sahn, A., von Vacano, C., and Kennedy, C. The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, pp. 83–94, Marseille, France, 2022. European Language Resources Association.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

A. Appendix

Our detailed code and dataset can be found in the GitHub repository at: https://github.com/yexin0720/5153_GP_Hate_Speech