Hate speech detection

Dzmitry Kurch, DSE

Text Mining and Sentiment Analysis project, August 2024

1. Introduction

The project focuses on hate speech detection, a crucial area in the field of text mining and sentiment analysis. Hate speech, which encompasses abusive, offensive, and harmful language, is a significant concern on online platforms. The ability to accurately detect and address hate speech is essential for creating safer and more inclusive online environments.

Hate speech detection is important for various applications. For instance, it can be used in controversial event extraction, where identifying hateful content can help in understanding and managing public discourse around sensitive topics. In AI chatterbots, ensuring that the system does not generate or endorse hate speech is critical for maintaining user trust and ethical standards. Content recommendation systems also benefit from hate speech detection by preventing the spread of harmful content. Additionally, in sentiment analysis, detecting hate speech helps in accurately gauging public sentiment, especially in discussions involving marginalized groups.

The data for this project is sourced from the Dynamically Generated Hate Speech Dataset https://github.com/bvidgen/Dynamically-Generated-Hate-Speech-Dataset, which was developed using a human and model-in-the-loop process. A reference paper can be found here https://www.google.com/url?q=https%3A%2F%2Faclanthology.org%2F2021.acl-long.132%2F. This approach aimed to address the limitations of existing hate speech detection models, which have struggled with performance, robustness, generalizability, and fairness. The dataset creation involved multiple rounds of data generation and model training. Initially, a classification model was trained using previously released hate speech datasets. Annotators were then tasked with finding content that would trick the model into making misclassifications. The misclassified content was used to train a new model in the next round, and this process was repeated for four rounds.

Each round involved a mix of original content and challenging perturbations, where the text was subtly altered to flip the label from 'Hate' to 'Not Hate' or vice versa. In the later rounds, annotators focused on creating more

adversarial and realistic content by closely examining real-world hate sites. This iterative process allowed for the development of a more robust and varied dataset, which is critical for training effective hate speech detection models.

Overall, this project aims to leverage the Dynamically Generated Hate Speech Dataset to explore and enhance the capabilities of hate speech detection models, ultimately contributing to safer and more inclusive online spaces.

2. Research question and methodology

The primary goal of this project is to develop a binary classification model capable of accurately detecting hate speech in text data. In formal terms, the problem can be defined as finding a function $f(x)$ that maps an input text x to a binary output y, where y is either 'Hate' or 'Not Hate'. The objective is to maximize the accuracy of the function $f(x)$, ensuring that the model correctly classifies as much text as possible.

In addition to building a high-performing hate speech detection model, this project also aims to compare the effectiveness of different modeling approaches. Specifically, we will evaluate classical machine learning models, such as Logistic Regression, using text embeddings derived from traditional methods like TF-IDF (Term Frequency-Inverse Document Frequency) and Word2Vec, against more advanced deep learning models, particularly pre-trained BERT-like models.

The performance of each model will be evaluated based on accuracy, and we will compare how well the traditional methods stack up against the more sophisticated deep learning approach. Through this comparison, we aim to understand the strengths and limitations of each approach and determine which method is most effective for hate speech detection.

3. Experiment results

In this section, we present the experimental results obtained from the various models tested during the project.

3.1 Data Preprocessing

Before feeding the text data into the models, we performed standard preprocessing steps to clean and normalize the data. This included:

- *Cleaning*: Removing special characters, punctuation, and other non-textual elements that could introduce noise into the models.
- *Normalization*: Converting all text to lowercase to ensure uniformity and reduce the dimensionality of the text data.

- *Lemmatization*: Reducing words to their base or root form, which helps in capturing the semantic meaning and reduces the vocabulary size.

These preprocessing steps helped to ensure that the text data was in an optimal format for model training and testing.

3.2 Data Analysis

To better understand the dataset, we conducted an exploratory analysis, including the creation of word clouds for the 'Hate' and 'Not Hate' classes. This analysis revealed that the dataset is quite challenging, with many misleading examples.

Figure 1 - NON-HATE text wordcloud

Figure 2 - HATE text wordcloud

As you can see words related to gender, race, and swearing appeared in both classes, making it difficult for classical machine learning models to differentiate between hate and non-hate speech. This is because traditional models often rely on individual tokens, whereas the context and sequence of tokens are critical in accurately classifying hate speech.

3.3 Model Performance

We experimented with both classical machine learning models and deep learning approaches to tackle the problem of hate speech detection.

- *Logistic Regression with TF-IDF*: The best-performing classical model was Logistic Regression using TF-IDF for text embedding. This model achieved a **66% accuracy** on the test set. Given the complexity of the dataset, this is a reasonable result, although it highlights the challenges posed by the dataset's misleading examples.
- *Logistic Regression with Word2Vec:* When using Word2Vec embeddings, the Logistic Regression model performed slightly worse than with TF-IDF,

showing the difficulty in capturing the necessary context for accurate classification using traditional methods.

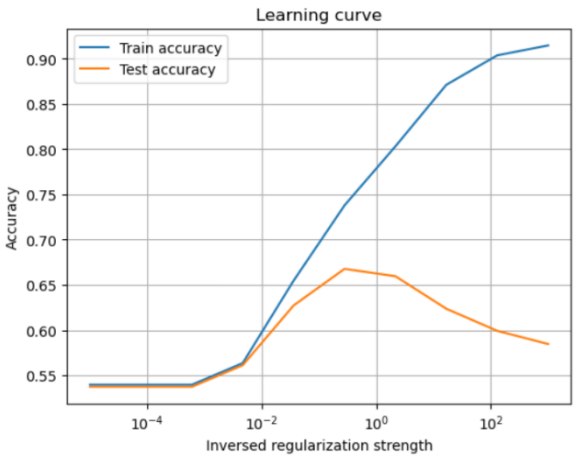


Figure 3 - Logistic Regression with TF-IDF learning curve

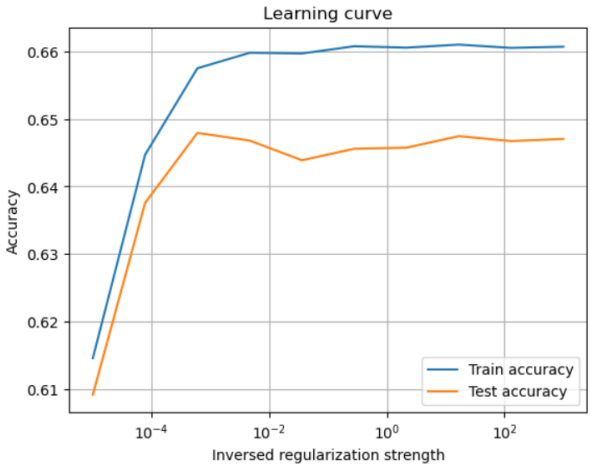


Figure 4 - Logistic Regression with Word2Vec learning curve

- *Fine-tuning ALBERT*: We then fine-tuned the albert-base-v2 model using the AlbertForSequenceClassification architecture. This deep learning model showed significant improvement over the classical approaches, achieving **77% accuracy** on the test set after just two epochs of training. This result underscores the advantage of using advanced deep learning models, which are better equipped to handle the complexities of natural language, such as context and token sequences.

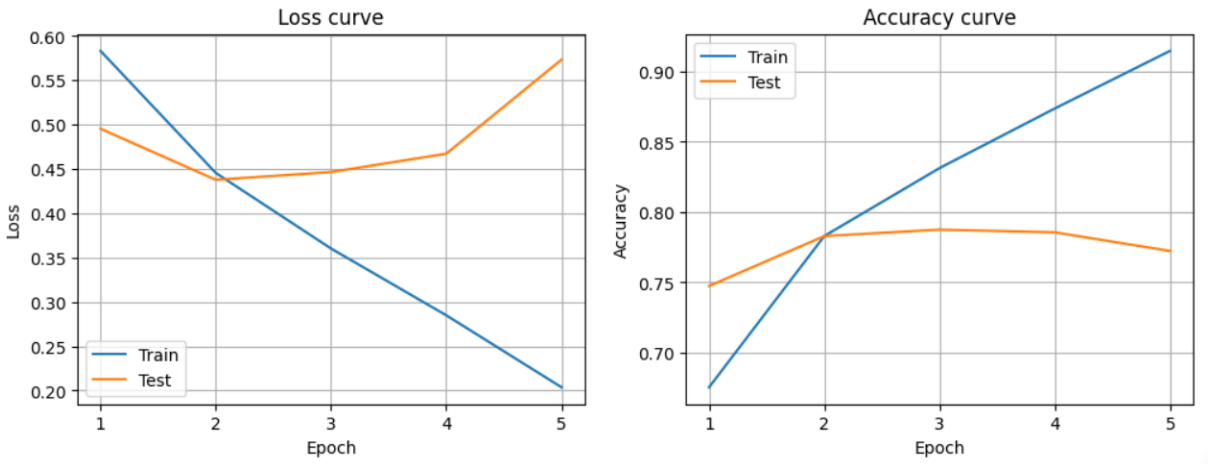
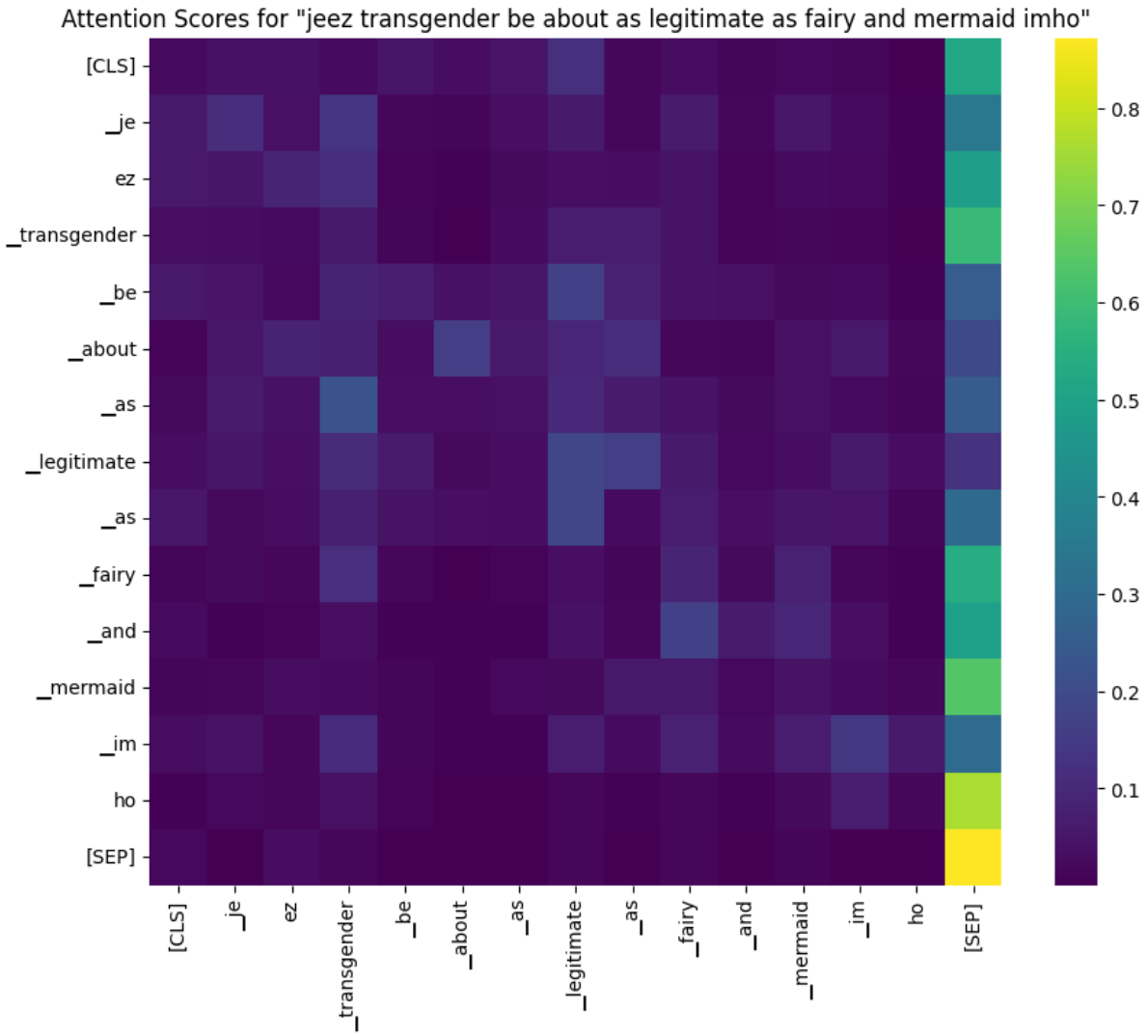


Figure 4 - ALBERT learning curve

3.4 Model Interpretation

To gain insights into how the albert-base-v2 model makes decisions, we analyzed the Attention matrices generated by the model for some input sentences. The Attention mechanism in Transformer-based models like ALBERT allows us to see which words or phrases the model is focusing on when making predictions. This analysis revealed that the model effectively captures the context in which

potentially harmful words are used, allowing it to differentiate between hate speech and non-hate speech more accurately than traditional models.



Overall, the experimental results demonstrate that while classical machine learning models can achieve moderate success in hate speech detection, deep learning approaches like ALBERT offer significantly better performance, especially on complex datasets with nuanced examples.

4. Concluding Remarks

In this project, we set out to develop a model for hate speech detection and compare the effectiveness of classical machine learning approaches against state-of-the-art deep learning models. The results of our experiments clearly show that while traditional models like Logistic Regression can perform reasonably well, they struggle with the complexities inherent in hate speech detection, particularly

when the dataset contains misleading examples where context and token sequence play a crucial role.

While the results from this project are promising, there is still room for further improvement. Future work could explore the following areas:

- **Data Augmentation**: Expanding the dataset with more diverse and challenging examples could help improve model robustness, especially in handling rare or subtle forms of hate speech.
- **Model Ensemble**: Combining the strengths of different models, such as an ensemble of classical and deep learning approaches, might yield even better performance.
- **Explainability**: Enhancing the interpretability of deep learning models through more advanced techniques could help in understanding and mitigating any biases or unfairness in the model's predictions.
- **Cross-domain Evaluation**: Testing the model on different datasets or across various platforms could provide insights into its generalizability and help refine it for broader applications.

In conclusion, this project underscores the importance of using advanced modeling techniques in hate speech detection and sets the stage for future research aimed at building even more effective and fair systems for maintaining safer online communities.