UNIVERSITÀ DEGLI STUDI DI MILANO

Text mining and Sentiment Analysis
Project: Hate Speech Detection (P7)

Ekaterina Sergeeva
ekaterina.sergeeva@studenti.unimi.it

December 2024

Contents

1 Introduction

Hate speech detection is a critical task in the field of text mining and sentiment analysis. It involves the identification of abusive, offensive, and harmful language, particularly on online platforms, where such content can contribute to toxic environments. This project focuses on classifying text as offensive (e.g., racist, sexist, or other hate speech) or non-offensive, while also exploring relevant terminology and linguistic patterns associated with hate speech.

The ability to accurately detect and address hate speech has widespread applications, such as enhancing content moderation systems, improving AI-powered conversational agents, and enabling ethical content recommendation systems. For instance, in controversial event extraction, hate speech detection can aid in understanding public discourse around sensitive topics. In AI chatterbots, it ensures that the systems avoid generating or endorsing hateful content, fostering user trust. Similarly, sentiment analysis can benefit from hate speech detection by providing a more accurate understanding of public sentiment, especially in contexts involving marginalized groups.

This project uses the **Dynamically Generated Hate Speech Dataset** to train and evaluate models for hate speech detection. This dataset was created through a human and model-in-the-loop process, addressing limitations in traditional datasets, such as biases, lack of robustness, and limited generalizability. The iterative creation process involved training models on existing hate speech datasets and challenging them with adversarial examples crafted by human annotators. The misclassified examples were used to retrain the models, resulting in a robust dataset that better captures the complexities of real-world hate speech.

This project aims to develop a binary classification model for detecting hate speech in text, mapping input text x to a binary label y (y = "Hate" or y = "Not Hate"). The goal is to maximize accuracy in distinguishing hateful language from neutral or positive language. While this project focuses on English, the methodology can be adapted to analyze hate speech in other languages, such as Italian, facilitating a comparative study of linguistic differences in hate speech across cultures.

By leveraging both traditional machine learning techniques (like Logistic Regression with TF-IDF and Word2Vec embeddings) and advanced transformer-based models (such as BERT), this project aims to explore the challenges and opportunities in hate speech detection. Additionally, it attempts to provide explanations for model decisions, contributing to the broader goal of creating safer and more inclusive digital spaces.

2 Experiment results

2.1 Data Preprocessing

A To prepare the dataset for model training, the following preprocessing steps were applied:

- **Cleaning**: Removed URLs, special characters, and punctuation to minimize noise.
- **Normalization**: Converted all text to lowercase for consistency and to reduce variability in the dataset.
- **Lemmatization**: Reduced words to their base forms to standardize the vocabulary and capture the semantic meaning of text.

These steps ensured that the data was in an optimal and uniform format, improving the quality of input for the classification models.

2.2 Data Analysis

An Exploratory analysis was conducted to better understand the dataset. This included creating a token distribution histogram and word clouds for the 'Hate' and 'Not Hate' classes.

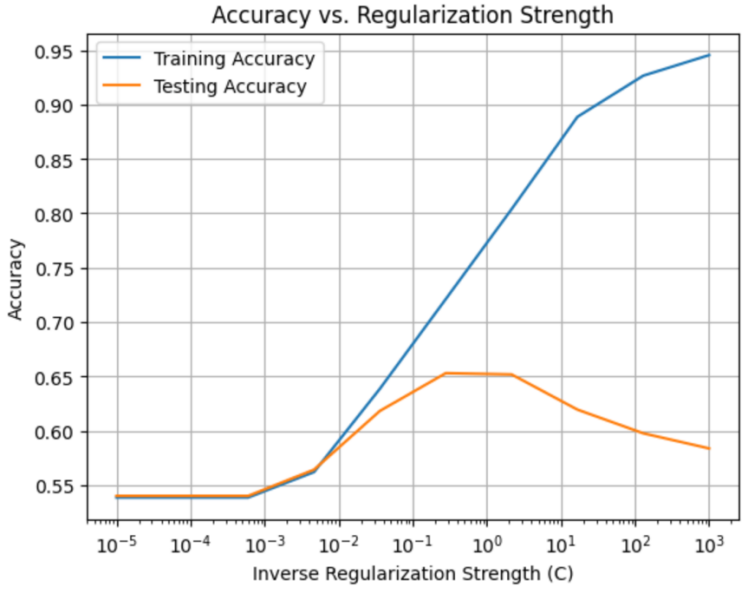
The analysis highlighted challenges in distinguishing hate speech from non-hate text due to overlapping vocabulary. For instance, words related to gender, race, and profanity appeared frequently in both classes, emphasizing the importance of considering the context rather than isolated words. This reinforces the need for advanced models like **BERT**, which take token sequence and context into account, unlike traditional models that rely solely on individual token importance.

These observations helped shape the modeling strategy by focusing on contextual approaches for better classification.

2.3 Model Performance: TF-IDF and Word2Vec

Logistic Regression with TF-IDF

The first approach involved using Logistic Regression with TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction. This classical machine learning method was evaluated across a range of regularization strengths, as illustrated in the "Accuracy vs. Regularization Strength" plot.



Training Results

The model demonstrated strong performance on the training data at higher inverse regularization strength values (C > 10), achieving a training accuracy close to 95%. However, this resulted in overfitting, as indicated by a significant gap between training and test accuracy.

Testing Results

The best test accuracy of approximately 64% was achieved with a regularization strength of C = 0.1. This value provided a balanced trade-off between underfitting and overfitting, ensuring better generalization to unseen data. Despite this, the test accuracy reflects the inherent challenges of the dataset, including overlapping vocabulary between hate and non-hate speech and nuanced contexts that are difficult to capture with traditional methods.

Observations from Regularization Strength

1. **Under-regularization (High C)**

   As C increased, the model began overfitting to the training data. While the training accuracy improved significantly, test accuracy showed a marked

decline, highlighting poor generalization.

2. **Optimal Regularization (C = 0.1)**

   At C = 0.1, the model achieved a balance between training and test accuracy. This point represents the best trade-off between bias and variance, ensuring the model is neither overly simplistic nor too closely fitted to the training data.

3. **Over-regularization (Low C)**
   At lower C values, the model struggled to capture meaningful patterns in the data. Both training and testing performance deteriorated, as the model was too constrained to learn the complex relationships within the dataset.

Classification Report

The classification report for the model provided the following key insights:

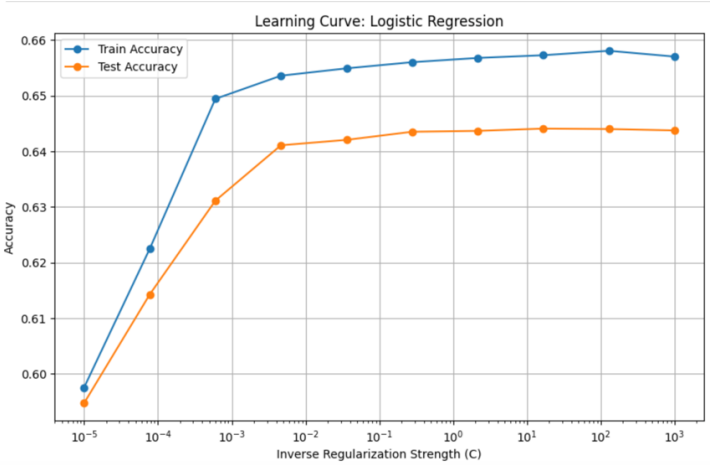
- Precision, recall, and F1-scores hovered around 0.64 for both 'Hate' and 'Not Hate' classes.

- The model showed a stronger recall for the 'Hate' class (0.76), meaning it was more effective at identifying hate speech instances. However, this came at the cost of a lower recall for the 'Not Hate' class (0.49), resulting in a higher rate of false positives.

Logistic Regression with Word2Vec

Word2Vec Embeddings with Logistic Regression

The second modeling approach leveraged Word2Vec embeddings as features for Logistic Regression. Word2Vec captures the semantic meaning of words by representing them as continuous vectors, making it a powerful feature extraction technique for text data. Similar to the TF-IDF approach, the model was evaluated across a range of regularization strengths, as depicted in the "Learning Curve: Logistic Regression" plot for Word2Vec.



Training Results

The model demonstrated consistent behavior during training, with training accuracy steadily improving as C increased (lower regularization). The highest training accuracy reached approximately 0.67 at C=1000. However, this also indicated a tendency toward overfitting, as evidenced by a widening gap between training and test accuracy at higher C values.

Testing Results

The best test accuracy of 63.5% was achieved with a regularization strength of C=0.001. This choice of C provided the optimal balance, mitigating overfitting while maintaining sufficient flexibility for the model to learn meaningful patterns in the data.

Observations from Regularization Strength

1. **Under-regularization (High C)**

   When C was too high, the model overfit the training data, leading to an increase in training accuracy but a corresponding decline in test performance.

2. **Optimal Regularization (C=0.001)**

   At C = 0.001, the model achieved a balance between bias and variance, with training and testing accuracies closely aligned, reflecting the model's ability to generalize well.

3. **Over-regularization (Low C)**

   With lower values of C, the model was overly constrained, leading to suboptimal performance on both training and testing data.

Classification Report

The classification report provided insights into the model's performance across the two classes ('Hate' and 'Not Hate'):

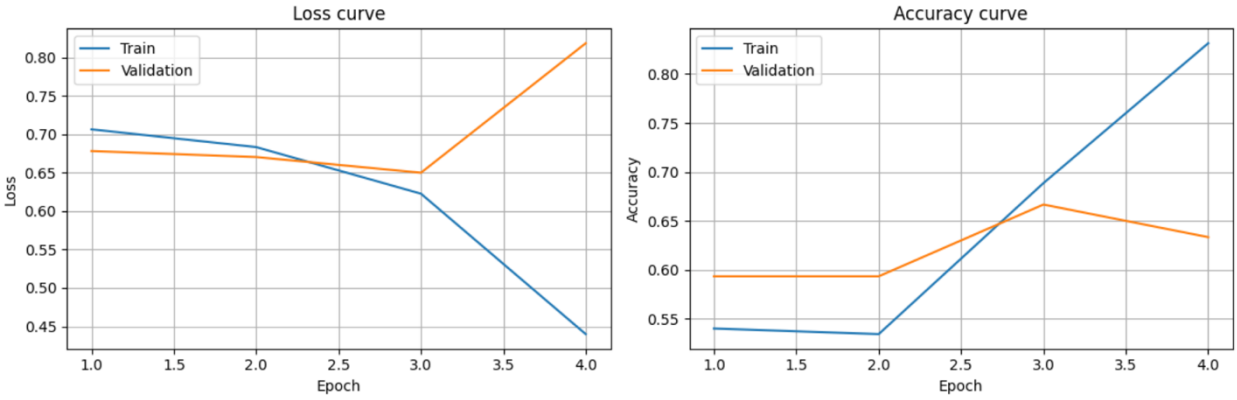
- **Precision, Recall, and F1-scores:** Both classes achieved balanced metrics, with precision, recall, and F1-scores averaging around 0.63.
- **Recall Disparity:** The model exhibited a slightly higher recall for the 'Hate' class (0.72) compared to the 'Not Hate' class (0.54). This indicates a better ability to identify hate speech instances but at the cost of generating more false positives for 'Not Hate'.

3 Model Performance: BERT-based Model

The third approach employed a pre-trained BERT model (Bidirectional Encoder Representations from Transformers) fine-tuned on the hate speech dataset. BERT is a state-of-the-art transformer-based architecture that excels at capturing contextual information, making it particularly suited for nuanced tasks like hate speech detection.

3.1 Training and Validation Results

The training and validation performance of the BERT model is depicted in the "Loss Curve" and "Accuracy Curve" plots.



- **Loss Trends**:
  The training loss steadily decreased across all four epochs, indicating that the model successfully learned patterns in the training data. However, the validation loss showed an initial decline before sharply increasing in the final epoch. This suggests overfitting as the model began to memorize training data rather than generalizing to unseen examples.
- **Accuracy Trends**:
  Training accuracy improved significantly, reaching 83.1% by the fourth epoch. In contrast, validation accuracy peaked at 66.7% during the third epoch before declining in the final epoch, further confirming signs of overfitting.

Classification Report

Evaluation of the fine-tuned BERT model on the test dataset produced the following results:

- **Overall Accuracy**: 63%
- **Class-wise Performance**:
  - 'Not Hate': Precision = 0.52, Recall = 0.90, F1-score = 0.66
  - 'Hate': Precision = 0.88, Recall = 0.46, F1-score = 0.60

The results indicate a trade-off between precision and recall for the two classes. While the model was more effective at identifying 'Not Hate' instances (high recall), it struggled with false negatives for the 'Hate' class, lowering its recall in that category.

Observations

1. **Strengths**:
    - BERT's contextual understanding enabled it to outperform traditional models in capturing the nuances of hate speech.
    - The ability to leverage pre-trained knowledge made the model robust against noisy and ambiguous examples.

2. **Limitations**:
    - The drop in validation performance during later epochs underscores the importance of careful hyperparameter tuning (e.g., regularization, dropout rates) to mitigate overfitting.
    - The model's recall imbalance indicates it struggled to achieve consistent performance across both classes, likely due to dataset complexity and class overlap.

3.2 Model Output Analysis

To evaluate the practical performance of the fine-tuned BERT model, a specific text sample was analyzed. The goal was to observe how the model processes real-world input and quantifies its classification probabilities.

Text Sample Evaluation

The selected from the database sample for evaluation was: "**Arabs are not welcome.**"

This text was tokenized using the BERT tokenizer, which splits the sentence into subword units and converts them into input IDs suitable for the model. After tokenization, the input was passed through the fine-tuned model in evaluation mode, and raw logits (pre-activation scores) were obtained. These logits were then converted to probabilities using the sigmoid function, which maps the values to a range between 0 and 1.

Results

The model produced the following output probabilities:

- **Probability of HATE:** 0.5549
- **Probability of NON-HATE:** 0.3508

This indicates that the model classified the input text as **HATE** with a moderate confidence of approximately 55.49%, while the probability of the text being **NON-HATE** was relatively lower at 35.08%.

Classification Confidence:

The model's classification of the text as "HATE" aligns with the apparent sentiment of the input, which contains exclusionary and hostile language. However, the confidence level (55.49%) is not overwhelmingly strong, suggesting that the model may have encountered ambiguity in the context or phrasing of the input.

3.3 Conclusion

The model successfully identified the given input as hate speech, demonstrating its ability to generalize to unseen text. However, the moderate confidence underscores the need for further enhancements, such as training on more diverse examples or leveraging ensemble approaches to boost performance. This case study highlights the practical application of the BERT model in hate speech detection and its potential for real-world use cases.

References

[1] Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021, May). Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 17, pp. 14867-14875).

[2] Arango, A., Pérez, J., & Poblete, B. (2019, July). Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 45-54).

[3] Fortuna, P.; and Nunes, S. 2018. A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR) 51(4): 85.