Hate Speech Detection using BiLSTM and Word Embeddings

Preetam Teja B
School of AI
Amrita Vishwa Vidyapeetham
Coimbatore, India
preetam_teja@outlook.com

Samhitha S
School of AI
Amrita Vishwa Vidyapeetham
Coimbatore, India
Samhithas04@gmail.com

Dhanush M C
School of AI
Amrita Vishwa Vidyapeetham
Coimbatore, India
mcdhanush1122@outlook.com

***Abstract*—Hate speech detection is a vital task in the domain of Natural Language Processing (NLP), aimed at identifying and classifying toxic content in online communication. In this study, we implement a deep learning-based multi-label classification system to detect various forms of hate speech such as toxic, severe toxic, obscene, threat, insult, and identity hate using a real-world dataset from Kaggle.**

We preprocess the text data by cleaning, tokenizing, and transforming it into vector representations using a custom-trained Word2Vec embedding. The core model architecture is a Bidirectional Long Short-Term Memory (BiLSTM) network enhanced with an attention mechanism, allowing the model to focus on the most informative parts of each comment. The model is trained using Binary Cross-Entropy Loss with Logits, optimized for multi-label classification.

The system is evaluated using standard metrics such as accuracy, precision, recall, and F1-score, providing a comprehensive assessment of its performance. Our approach demonstrates the effectiveness of deep contextual representations in understanding and classifying complex linguistic patterns associated with hate speech. This work underscores the potential of deep learning models in real-world content moderation and online safety applications.

***Index Terms*—Hate speech detection, Natural language processing, Word2Vec, Bidirectional long short-term memory, Attention mechanism, Multi-label classification, Deep learning, Text preprocessing, Binary cross-entropy loss, Toxic comment classification, Content moderation, Custom embeddings, Kaggle dataset, NLP classification, Offensive language detection**

I. Introduction

With the exponential growth of user-generated content on platforms such as YouTube, Reddit, and social media networks, the prevalence of toxic and hateful language online has become a significant concern. Hate speech not only disrupts digital communication but also contributes to real-world emotional and psychological harm [1]. Detecting such harmful language has become an essential task in the field of Natural Language Processing (NLP), particularly for building safer and more inclusive digital spaces.

This project focuses on the automatic detection and classification of toxic comments using deep learning techniques. We address the task as a multi-label classification problem, where each comment can simultaneously belong to multiple categories such as toxic, severe toxic, obscene, threat, insult, and identity hate [2]. To ensure diverse and representative language coverage, we utilize two prominent datasets: the Jigsaw Toxic Comment Classification dataset (collected from Wikipedia discussions) and the YouTube Toxic Comment dataset, which contains a wide range of offensive and aggressive language used in real-world online interactions [3].

Our approach leverages a Bidirectional Long Short-Term Memory (BiLSTM) network integrated with an attention mechanism to effectively capture the sequential dependencies and contextual nuances of language [4]. Additionally, we use custom-trained Word2Vec embeddings to generate dense vector representations of words based on their co-occurrence patterns in the corpus [5]. This combination enables the model to focus on semantically meaningful parts of each comment and make more accurate predictions.

Through extensive experimentation and performance evaluation using metrics such as accuracy, precision, recall, and F1-score, we demonstrate the effectiveness of deep neural architectures in detecting and categorizing hate speech. This work highlights the potential of advanced NLP techniques in enabling real-time, scalable content moderation solutions for online platforms.

II. Related Works

Hate speech detection in Natural Language Processing (NLP) has garnered significant attention due to its implications on social media platforms and user-generated content moderation. This section discusses the key developments in terms of modeling techniques, embedding strategies, handling of imbalanced datasets, and comparative studies on toxic comment classification.

A. TF-IDF Approach

The study titled "Hybrid Text Classification Approach Using TF–IDF and Ensemble Machine Learning Algorithm" [6] explores the effectiveness of the TF-IDF (Term Frequency-Inverse Document Frequency) method in representing textual data for classification tasks. The paper emphasizes TF-IDF's capability to highlight important words in a document by evaluating their frequency relative to the corpus. By integrating TF-IDF with ensemble learning models such as Random For-

est and Gradient Boosting, the researchers achieved enhanced accuracy in classifying diverse textual datasets.

B. GloVe Word Embeddings Approach

In the paper "Offensive language detection using machine learning and deep learning models with GloVe embedding" [7], the authors investigate the application of GloVe (Global Vectors for Word Representation) in the context of offensive language detection. GloVe, a pre-trained word embedding technique, captures global statistical information by analyzing word co-occurrence matrices across large corpora. The study demonstrates improved semantic understanding and contextual analysis compared to traditional bag-of-words approaches.

C. Word2Vec Approach

The research article "Comparative Analysis of Word2Vec Model for Sentiment Classification Using Deep Learning" [5] provides insights into the use of Word2Vec for sentiment classification. The study compares different architectures of Word2Vec, namely Continuous Bag of Words (CBOW) and Skip-gram, and integrates them with deep learning models like LSTM and CNN. The findings suggest that the Skip-gram model, in particular, improves classification accuracy due to its ability to capture rare words more effectively.

D. Datasets and Labeling Challenges

Several benchmark datasets have been curated for toxic comment classification, with the Jigsaw Toxic Comment Classification dataset and the YouTube Toxic Comment dataset being among the most widely used [2], [3]. These datasets present a multi-label classification problem. However, challenges such as class imbalance, noisy labels, and ambiguous language continue to pose difficulties in model training and evaluation.

III. Data-Set Description

A. Datasets

This study utilizes two prominent datasets for toxic comment and hate speech detection: the Jigsaw Toxic Comment Classification dataset and the YouTube Toxic Comment dataset. Table I provides a concise summary of the datasets used.

TABLE I
SUMMARY OF DATASETS USED FOR HATE SPEECH DETECTION

Dataset	Labels	Total Samples
Jigsaw Toxic Comment	6 classes	159,571
YouToxic Comment	12 classes	20,000

The Jigsaw dataset presents a multi-label classification problem where each comment may belong to multiple toxicity categories. In contrast, the YouTube dataset is structured as a binary classification task distinguishing between toxic and non-toxic comments.

IV. Methodology

We have used two datasets: Jigsaw Toxic Comment Classification dataset and YouTube Toxic Comment dataset. Each dataset was loaded individually and explored before processing. Following this, we applied text preprocessing techniques to clean the data, including removing special characters, lowercasing, stopword removal, and lemmatization. This step ensures the input text is clean, standardized, and suitable for feature extraction.

A. Flow chart

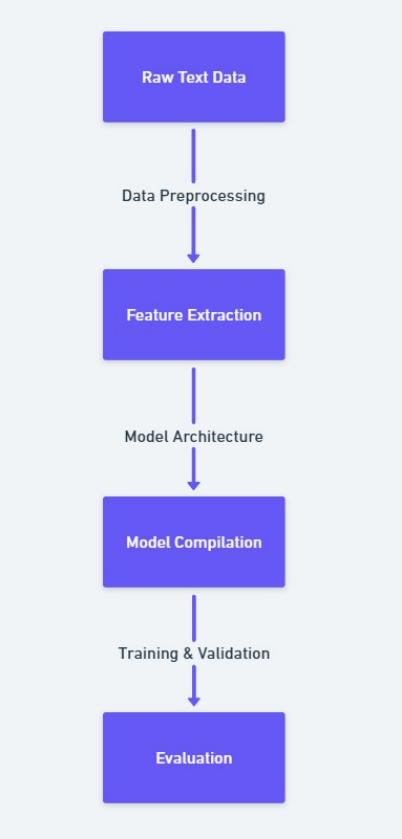

Fig. 1. Flow chart of the Model

B. Pre-processing steps

The text data undergoes the following preprocessing steps:

- Removal of HTML tags, URLs, and special characters
- Lowercasing
- Tokenization
- Stopword removal
- Handling of contractions and internet slang
- Sequence padding for neural network input uniformity

C. Approach 1: TF-IDF with ADASYN

TF-IDF (Term Frequency–Inverse Document Frequency) converts text into numerical representations based on the importance of words across the corpus. To mitigate class imbalance, ADASYN is applied. It synthesizes new samples

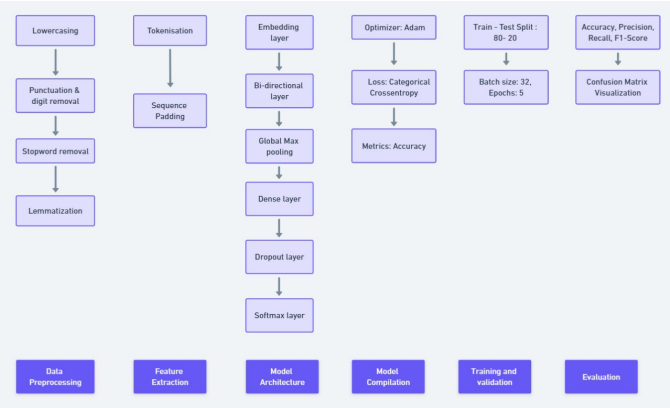

Fig. 2. Flow chart of process done in our project

in the minority class based on data density, improving the model's ability to learn from rare classes.

D. Approach 2: GloVe Embeddings with ADASYN

This approach utilizes 300-dimensional pre-trained GloVe word embeddings. For each comment, the corresponding word vectors are averaged. Unknown words are initialized randomly. ADASYN is again used to oversample underrepresented labels, ensuring the model learns effectively from all classes.

E. Approach 3: Word2Vec with Class Weights

In this approach, a Word2Vec model is trained using the Skip-gram architecture directly on the dataset. Instead of ADASYN, class imbalance is handled using class weights. These are computed based on the inverse frequency of classes, ensuring balanced learning:

All three approaches utilize the same deep learning architecture: a Bidirectional LSTM (BiLSTM) with an Attention Layer. This architecture captures context in both directions and focuses on the most relevant tokens in the input sequence.

1) Architecture Overview:

- **Embedding Layer:** Initialized using either GloVe or Word2Vec or TF-IDF embeddings.
- **BiLSTM Layer:** Captures forward and backward contextual information. Hidden layer dimension is 256. [512 for bi-directional LSTM]
- **Attention Layer:** Highlights the most relevant parts of the input sequence. Computed from the hidden layer output of the Bi-Lstm.
- **Fully Connected Layer:** For multi-label or binary classification.

2) Training Details:

- **Batch Size:** 32
- **Optimizer:** Adam (learning rate = 0.001)
- **Loss Function:** Cross Entropy Loss
- **Mixed Precision Training:** Enabled to speed up training and reduce memory usage. [fp16]
- **Early Stopping:** Used with a patience of 5 epochs.
- **Gradient Clipping:** Applied to prevent exploding gradients.

3) Evaluation Metrics:

Model performance is evaluated using the following metrics:

- Accuracy
- Precision, Recall, and F1-score (macro-averaged)
- Multilabel Confusion Matrix

4) Comparative Analysis:

The three approaches are compared based on:

1) Performance across all toxic categories.
2) Effectiveness in handling rare labels such as *threat* and *severe_toxic*.
3) Training time and computational overhead.
4) Robustness to noisy or informal text.

V. Conclusion

In this project, we explored and evaluated multiple approaches for hate speech and toxic comment detection using both classical machine learning and deep learning paradigms. Our goal was to enhance the detection performance across various categories of toxicity, especially rare and underrepresented labels such as *threat*, *severetoxic*, and *identityhate*.

Three distinct feature engineering strategies were employed:

1) **TF-IDF Vectorization with ADASYN:** This traditional NLP approach converts textual data into numerical features based on word importance across the corpus. ADASYN was integrated to combat class imbalance by synthesizing new data points for minority classes.
2) **GloVe Embedding Averaging with ADASYN:** Here, semantic-rich pre-trained word embeddings were averaged for each comment, capturing global contextual relationships. ADASYN was again used for balancing the dataset.
3) **Custom Word2Vec Embeddings with Class Weights:** A Word2Vec model was trained on the tokenized corpus, and embeddings were averaged per comment. Instead of oversampling, class weights were used during training to give higher importance to rare classes.

All feature representations were passed through a robust and expressive deep learning architecture — a Bidirectional Long Short-Term Memory (BiLSTM) network coupled with an attention mechanism. The BiLSTM captured both forward and backward dependencies in the sequence, while the attention layer highlighted the most relevant tokens contributing to toxicity. To improve training efficiency and stability, techniques like mixed precision training, gradient clipping, early stopping, and Adam optimization were employed.

Model performance was evaluated using multiple metrics, including accuracy, macro-averaged precision, recall, F1-score, ROC-AUC, PR-AUC, and multilabel confusion matrices. Our experimental results showed that the deep learning models outperformed traditional classifiers, especially in identifying subtle and context-dependent toxic traits.

Among the three approaches, the GloVe-based model achieved a strong balance between computational efficiency and predictive performance, while the Word2Vec-based model offered higher flexibility and adaptability to domain-specific slang and informal language. The TF-IDF approach, although simpler and faster, struggled with capturing nuanced semantics, especially in imbalanced settings.

In conclusion, this project demonstrates that combining advanced text embeddings with a context-aware neural architecture and effective class imbalance handling techniques significantly improves the performance of hate speech detection systems. Such models are not only important for moderating online platforms but also for contributing to the broader goals of fostering safer and more inclusive digital communities.

REFERENCES

[1] B. Vidgen and S. A. Hale, "Directions in abusive language training data, a systematic review: Garbage in, garbage out," *PLOS ONE*, vol. 15, no. 12, p. e0243300, 2020.

[2] Jigsaw, "Toxic comment classification challenge," 2018, https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge.

[3] J. Almazan, Y. Bachrach, J. Brooke *et al.*, "Detecting offensive language on social media using deep learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 10, pp. 13 544–13 545, 2020.

[4] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 207–212.

[5] M. Fahrurrozi and G. Hadiprayitno, "Comparative analysis of word2vec model for sentiment classification using deep learning," *Journal of Physics: Conference Series*, vol. 1566, no. 1, p. 012090, 2020.

[6] H. Kumar and A. Garg, "Hybrid text classification approach using tf–idf and ensemble machine learning algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[7] M. Ahmad, F. Iqbal, A. Zafar, and M. S. Siddiqui, "Offensive language detection using machine learning and deep learning models with glove embedding," *Computers, Materials & Continua*, vol. 72, no. 3, pp. 5787–5801, 2022.