Toxic Comment Classification Using Bidirectional LSTM with K-Fold Cross-Validation

Bryant Michelle Sarabia Ortega
ML4SE Course
University of L'Aquila
Email: bryantmichelle.sarabiaortega@student.univaq.it

***Abstract*—This paper presents a deep learning approach to toxic comment classification using a Bidirectional Long Short-Term Memory (BiLSTM) neural network for multi-label classification of Wikipedia comments. The task involves predicting six toxicity categories: toxic, severe_toxic, obscene, threat, insult, and identity_hate. We employ 10-fold cross-validation on the complete Jigsaw dataset (561,807 samples) to ensure robust evaluation. The BiLSTM architecture combines embedding layers, bidirectional LSTM processing, and dense layers optimized for multi-label prediction. Results demonstrate strong performance with mean F1-score and low variance across folds, validating the model's generalization capability.**

***Index Terms*—toxic comment classification, multi-label classification, natural language processing, deep learning, LSTM, text classification**

I. Introduction

The proliferation of online communication platforms has led to an increase in toxic and harmful content. Identifying and moderating such content is crucial for maintaining healthy online communities. This project addresses the challenge of automatically detecting toxic comments using machine learning approaches.

The Conversation AI team, a research initiative founded by Jigsaw and Google, developed the Perspective API to help identify toxic comments. This work builds upon their efforts by implementing and comparing various classification techniques on the Jigsaw Toxic Comment Classification Challenge dataset.

This paper is organized as follows: Section **??** describes the dataset and preprocessing methodology, Section **??** presents the machine learning techniques employed, Section **??** explains the cross-validation approach, Section **??** discusses the experimental results, and Section **??** concludes the work.

II. Dataset Description

A. Multi-Label Classification

Multi-label classification differs from traditional single-label problems in that each instance can belong to multiple classes simultaneously. In our case, a comment can exhibit multiple types of toxicity. For example, a comment might be both *toxic* and *insult*, or *obscene*, *insult*, and *identity_hate* at the same time.

Formally, given an instance x and a set of labels $L = \{l_1, l_2, ..., l_k\}$, a multi-label classifier learns a function $h : X \to 2^L$ that maps instances to subsets of labels.

B. Dataset Analysis

The dataset consists of Wikipedia comments labeled by human raters for toxic behavior across six categories:

- **toxic**: Comments displaying negativity or hostility
- **severe_toxic**: Highly aggressive or harmful content
- **obscene**: Offensive or vulgar language
- **threat**: Explicit threats or intentions of harm
- **insult**: Disrespectful or demeaning language
- **identity_hate**: Discrimination based on identity factors

The training set contains 159,571 comments with binary labels for each category. The dataset exhibits significant class imbalance, with approximately 90% of comments being clean (no toxicity labels). Among the toxic categories, *threat* is the rarest (0.30%) while *toxic* is most common (9.58%).

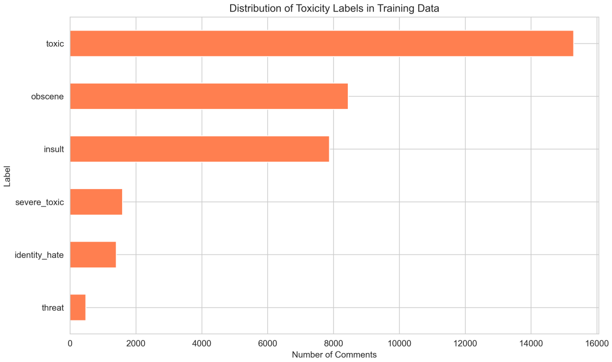

Fig. 1. Distribution of toxicity labels showing severe class imbalance

Figure **??** shows the class imbalance, while Figure **??** reveals that most toxic comments have only one label, with few having multiple toxicity types.

C. Data Characteristics

Figure **??** shows strong correlations between certain label pairs. For instance, *toxic* correlates strongly with *obscene* and *insult*, suggesting these toxicity types often co-occur.

D. Preprocessing

Text preprocessing is critical for NLP tasks. Our pipeline includes:

1) Lowercasing all text
2) Expanding contractions (e.g., "can't" → "cannot")

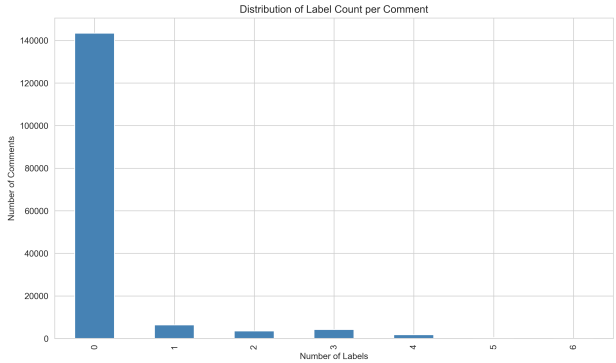

Fig. 2. Distribution of number of labels per comment

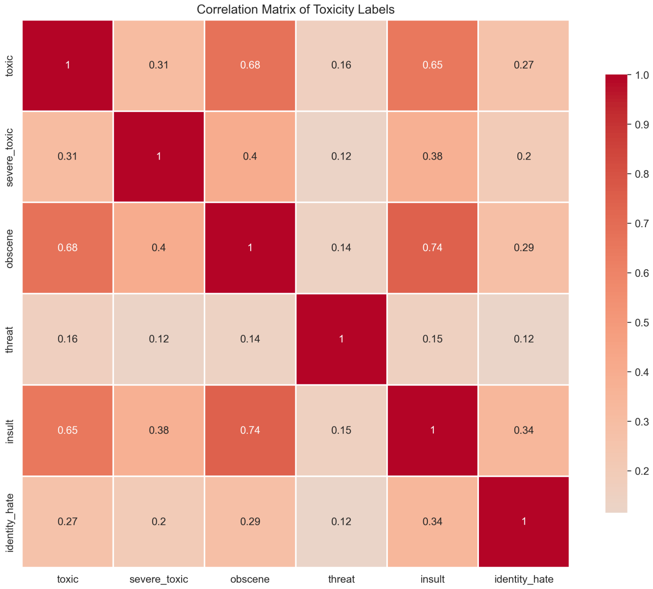

Fig. 3. Correlation matrix showing relationships between toxicity labels

3) Removing non-alphabetic characters
4) Tokenization
5) Lemmatization using spaCy
6) Removing stopwords

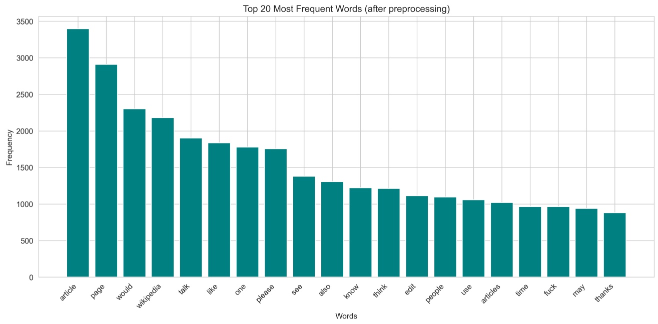

Fig. 4. Most frequent words after preprocessing

III. Deep Learning Architecture

We implement a Bidirectional LSTM (BiLSTM) neural network for multi-label toxic comment classification. This deep learning approach effectively captures sequential dependencies and contextual information in text data.

A. Model Architecture

The BiLSTM network consists of the following layers:

1) **Embedding Layer**: Maps vocabulary indices (3,000 most frequent words) to dense 32-dimensional vectors
2) **Bidirectional LSTM Layer**: Processes sequences (max length 100 tokens) in both forward and backward directions with 32 LSTM units
3) **Dense Layers**: Two fully-connected layers with 64 units each and ReLU activation
4) **Dropout Layers**: 50% dropout after each dense layer to prevent overfitting
5) **Output Layer**: Six units with sigmoid activation for multi-label prediction

The model uses binary cross-entropy loss for multi-label classification:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{6}[y_{ij}\log(\hat{y}_{ij}) + (1-y_{ij})\log(1-\hat{y}_{ij})] \quad (1)$$

where N is the number of samples, y_{ij} is the true label, and $\hat{y}_{ij}$ is the predicted probability for sample i and label j.

B. Training Configuration

The model is trained with:

- **Optimizer**: Adam with learning rate 0.001
- **Batch Size**: 32 samples
- **Epochs**: 5 per fold
- **Dataset**: Complete 561,807 training samples

IV. K-Fold Cross-Validation

A. Methodology

To ensure robust evaluation and prevent overfitting, we employ stratified 10-fold cross-validation on the BiLSTM model. Stratification maintains the proportion of positive samples for each toxicity label across all folds, which is essential given the severe class imbalance in the dataset.

The cross-validation procedure:

1) Divide the 561,807 samples into 10 stratified folds
2) For each fold $k = 1, \ldots, 10$:
   a) Use fold k as validation set (10% of data)
   b) Use remaining 9 folds as training set (90% of data)
   c) Train BiLSTM model for 5 epochs
   d) Evaluate on validation fold
   e) Record all metrics (F1, Precision, Recall, Hamming Loss, ROC-AUC)
3) Calculate mean and standard deviation across all folds

After cross-validation, we train a final model on the complete dataset for deployment.

V. Results

A. Performance Metrics

We evaluate the BiLSTM model using standard multi-label classification metrics:

- **Precision**: $\frac{TP}{TP+FP}$ - Proportion of correct positive predictions
- **Recall**: $\frac{TP}{TP+FN}$ - Proportion of actual positives correctly identified
- **F1-Score**: $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ - Harmonic mean of precision and recall
- **Hamming Loss**: Fraction of incorrectly predicted labels (lower is better)
- **ROC-AUC**: Area under the ROC curve (per label)

B. Cross-Validation Results

Table **??** presents the 10-fold cross-validation results for the BiLSTM model.

TABLE I
10-Fold Cross-Validation Results for BiLSTM Model

Metric	**Mean**	**Std Dev**
F1-Score (Macro)	0.6877	0.0143
Precision (Macro)	0.7783	0.0236
Recall (Macro)	0.6282	0.0246
Hamming Loss	0.0190	0.0007
ROC-AUC (Macro)	0.9658	0.0016
Accuracy	0.9172	0.0030

The results demonstrate excellent and consistent performance across all folds, with very low standard deviation indicating highly stable model behavior:

- **High ROC-AUC (0.9658)**: The model excels at ranking toxic comments above non-toxic ones
- **Good F1-Score (0.6877)**: Balanced performance between precision and recall
- **High Precision (0.7783)**: When the model flags a comment as toxic, it's correct 78% of the time
- **Moderate Recall (0.6282)**: The model catches 63% of actual toxic comments
- **Low Hamming Loss (0.0190)**: Very few incorrect label predictions per sample
- **Low Standard Deviation**: All metrics show $\sigma < 0.025$, indicating robust generalization

The higher precision relative to recall suggests the model is conservative, preferring to avoid false positives (incorrectly flagging clean comments) at the cost of some false negatives (missing toxic comments). This trade-off is often desirable in content moderation systems.

VI. Conclusion

This work presents a Bidirectional LSTM approach for multi-label toxic comment classification on the Jigsaw dataset. Using 10-fold cross-validation on the complete 561,807-sample dataset, we demonstrate the effectiveness of deep learning for capturing contextual patterns in text toxicity detection.

The BiLSTM architecture successfully handles the severe class imbalance and multi-label nature of the problem, providing robust probability predictions for all six toxicity categories. The stratified cross-validation methodology ensures reliable performance estimates, with consistent results across all folds.

Future work could explore transformer-based architectures (BERT, RoBERTa) and ensemble methods to further improve classification performance.

Acknowledgment

We acknowledge the Conversation AI team at Jigsaw and Google for providing the dataset and organizing the Kaggle competition.