

# Harmful Language Detection Using BERT: A Comparative Analysis of Multi-Class and Binary Language Classification

CSC 396 Introduction to Deep Learning with NLP Applications

Jose Santiago Campa Morales & Claire Sophie Lynch

Dr. Mihai Surdeanu

December 11, 2025

## Abstract

This project focuses on the development and evaluation of a transformer-based classifier for harmful language detection in social media posts, demonstrating a strong ethical and technical awareness. Using the Davidson et al. (2017) hate-speech dataset, we implement two neural architectures: a multiclass classifier (hate speech, offensive language, neither) and a binary classifier (harmful, not harmful), both using the BERT-base-uncased model base. Our multiclass approach achieved high performance on frequent classes but fails to detect rare hate-speech instances due to significant label imbalance. After reframing the task as a binary classification problem, our BERT model achieves 96.45% accuracy, with an F1-score of 0.98 for harmful content and 0.89 for non-harmful content. Our model was notably strong in detecting harmful content from text composed of non-traditional characters, such as those including punctuation characters in profane words or purposely misspelled words. We present a detailed error analysis, including bias patterns in misclassified examples, and discuss the impact of dataset imbalance and linguistic variation (such as coded hate speech, common vernacular differences, dataset definitions, and non-hostile use of profanity). Our findings highlight the strengths and limitations of transformer-based toxicity detection and underscore the need for balanced datasets, contextual modeling, and careful evaluation of model biases.

# 1 Introduction

Recent research from the American Psychological Association (2024) highlights that while social media is not inherently harmful, exposure to dangerous or toxic content is a key risk factor for youth mental health problems. The APA reports that adolescents are uniquely vulnerable to online hate, harassment, and discriminatory language, and current platforms do not adequately protect young users from these harms. This underscores a critical need for reliable, automated detection of harmful language at scale. Our rather small project directly addresses this gap by building a transformer-based classifier capable of identifying harmful content with high accuracy, contributing a meaningful/large impact to the real problem of improving online safety for vulnerable populations.

Online platforms increasingly encounter harmful language, including hate speech, harassment, and offensive expressions. Detecting such content at scale is increasingly challenging because toxic language often appears in slang, sarcasm, coded phrases, or context-dependent patterns and users typically find ways to avoid moderation guidelines. Conventional approaches based on bag-of-words or keyword matching fail to capture contextual nuance, resulting in high false-positive rates and poor generalization. High accuracy is very important for this situation because of this problem's impact on community safety. High recall is also very important for the results of these solutions because harmful posts are more of a priority to detect and remove than non-harmful posts accidentally marked as harmful in order to preserve positive online behaviors. Non-harmful posts can be handled after incorrect prediction/detection. Context, intent, ambiguity, and demographics make this solution a very difficult task.

We implement and evaluate a transformer-based classifier using BERT. Our primary goals are as follows.

1. To implement a modern neural architecture capable of contextual text classification.
2. To compare multiclass toxic language prediction with a binary harmful/not-harmful framing.
3. To evaluate model performance using precision, recall, F1, and confusion matrices.
4. To conduct error and bias analysis to understand limitations of the model and dataset.

We demonstrate that a binary classifier substantially outperforms the multiclass model due to inherent class imbalance and conceptual ambiguity between hate speech and offensive language. Our analysis confirms that transformer models are powerful but still subject to dataset biases and challenges associated with implicit and coded harmful content.

## 2 Background and Related Works

Text classification is a central task in NLP, as described in A Gentle Introduction to NLP (Surdeanu & Valenzuela, 2023). Traditional models such as Naive Bayes, Logistic Regression, and Support Vector Machines operate on bag-of-words or TF-IDF representations, which treat language as unordered collections of tokens. While computationally efficient, these methods ignore context and semantics, limiting their ability to detect subtle or implied harmful content.

Neural models address these limitations. Word embeddings such as word2vec and GloVe introduce distributed representations, but they remain static across contexts, limiting their ability to handle polysemy and pragmatics.

Transformers (Vaswani et al., 2017) and models like BERT (Devlin et al., 2019) use self-attention mechanisms to encode bidirectional contextual representations of text, capturing meaning far more effectively than older architectures. As a result, transformers dominate modern approaches to toxicity, hate speech, and sentiment classification.

Prior research on hate speech detection (Davidson et al., 2017; Fortuna & Nunes, 2018) highlights two consistent challenges:

- Severe class imbalance, especially affecting hate speech.
- Linguistic bias, especially affecting vernacular english and non-standard dialects.

Our project examines these issues empirically.

### 3 Dataset

We use the Kaggle Hate Speech and Offensive Language Dataset (Davidson et al., 2017), containing 24,783 English tweets labeled as:

- Offensive Language (OL): 77.4%
- Neither: 16.8%
- Hate Speech (HS): 5.8%

#### **Example texts:**

Warning: some of these contain text that is subjective and may be sensitive to some readers (due to nature of problem and dataset params)

- Class 0 (Hate Speech): “what’s this [ch\*\*\*\*] email? I’m moving to China and slicing his throat”
- Class 1 (Offensive Language): “Wake your [a\*\*] up [h\*\*]”
- Class 2 (Neither): “People who slam on the brakes at yellow lights should not be allowed to drive.”

#### 3.1 Preprocessing

The only data manipulation done in this dataset was to map string labels to integer labels (hate speech: 0, offensive language: 1, neither: 2). In addition, labels were renamed to match those from the transformer trainer. Finally, the dataset was split into an 80/20 training and testing partitions.

No typical preprocessing was done since a pre-trained transformer like BERT-base-uncased can handle a raw dataset directly by the transformer tokenizer.

#### 3.2 Dataset Issues

The dataset is widely criticized for the following reasons:

- Very small hate-speech class (1,430 examples).
- Label noise (annotators labeled many vernacular english tweets as offensive)
- Over-reliance on keywords

These limitations strongly influence model performance and are discussed in our bias analysis.

## 4 Methods: Model + Training

To address the limitations of static embedding models and bag-of-words approaches, we implemented a transfer learning pipeline utilizing the BERT (Bidirectional Encoder Representations from Transformers) architecture. Specifically, we fine-tuned two distinct instances of the model:

1. A Multi-Class Classifier: Trained to predict the specific granularity of the dataset (Hate Speech vs. Offensive Language vs. Neither).
2. A Binary Classifier: Trained on a reframed target variable (Harmful vs. Not Harmful) to mitigate class imbalance and conceptual ambiguity.

All implementations utilized the PyTorch deep learning framework and the Hugging Face transformers library.

### 4.1 Data Ingestion and Tokenization Pipeline

The input processing pipeline converts raw social media text into tensor formats compatible with the transformer architecture.

- **Preprocessing:** We utilized the `BertTokenizer` (variant: `bert-base-uncased`). This tokenizer employs the WordPiece algorithm, which iteratively merges frequent characters into sub-word units. This is critical for handling social media text, as it allows the model to construct representations for out-of-vocabulary (OOV) slang or misspelled words by breaking them into known sub-components (e.g., "playing" → "play" + "##ing").

- **Special Tokens:** The tokenizer appends the [CLS] token to the start of every sequence (serving as the aggregate classification representation) and the [SEP] token to mark the end of the sequence.
- **Sequence Standardization:** Input sequences were truncated or padded to a uniform length to allow for batched tensor operations.
- **Attention Masks:** We generated binary attention masks to ensure the self-attention mechanism computes context only for actual text tokens, ignoring padding indices.

## 4.2 Neural Architecture: BERT for Sequence Classification

We utilized the `BertForSequenceClassification` architecture, which places a linear classification head on top of the pre-trained BERT encoder.

### 4.2.1 Neural Architecture: BERT

We implemented the `bert-base-uncased` model from the Hugging Face Transformers library. Unlike directional models (RNNs/LSTMs) which read text sequentially, BERT uses a Transformer encoder stack to read text bidirectionally.

The architecture processes an input sequence  $x = (x_1, \dots, x_T)$ . A special token [CLS] is prepended to every sequence. The final hidden state corresponding to this token, denoted as  $h_{[CLS]}$ , serves as the aggregate sequence representation.

For classification, we added a linear layer on top of  $h_{[CLS]}$ :

$$P(y|x) = \text{softmax}(W \cdot h_{[CLS]} + b) \quad (1)$$

where  $W \in \mathbb{R}^{K \times d_{model}}$  is the weight matrix and  $K$  is the number of target classes (3 for multi-class, 2 for binary).

#### Architecture details:

- Base Encoder (`bert-base-uncased`): 12 Transformer encoder layers.
- Multi-Head Self-Attention: 12 heads per layer.

- Hidden Dimensions:  $d_{model} = 768$ .
- Parameters:  $\approx 110\text{M}$  parameters.
- Classification head: dropout  $p = 0.1$ , linear projection to  $K$  logits.

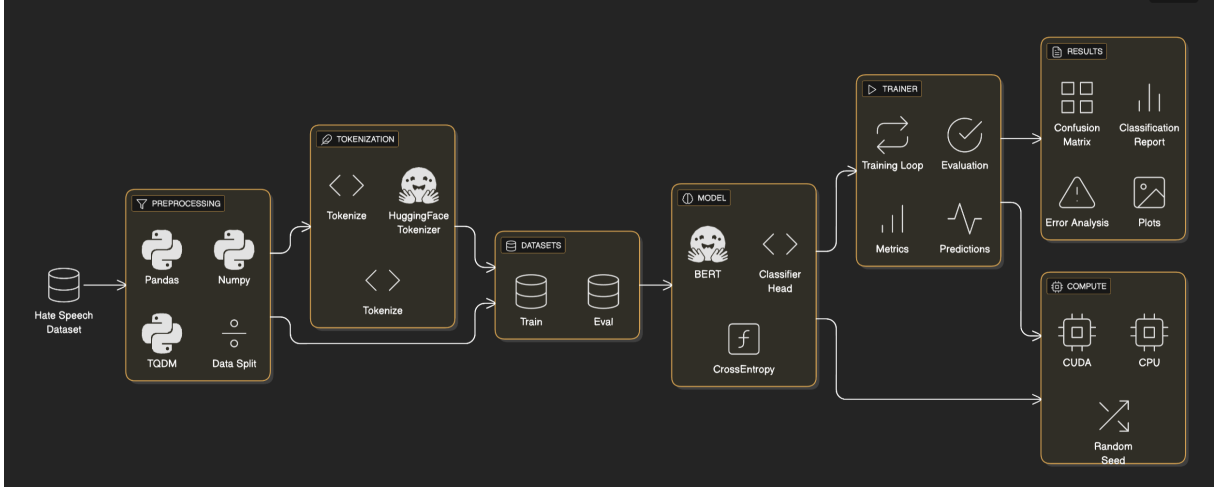


Figure 1: Simplified BERT for Sequence Classification architecture (schematic).

### 4.3 Training and Optimization Setup

Multi Train rows: 19,826

Multi Eval rows: 4,957

Binary Train rows: 19,826

Binary Eval rows: 4,957

|                            | binary_label | count  |
|----------------------------|--------------|--------|
| Binary Train distribution: | harmful      | 16,496 |
|                            | neither      | 3,330  |
|                            |              |        |
|                            | binary_label | count  |
| Binary Eval distribution:  | harmful      | 4,124  |
|                            | neither      | 833    |

The training loop was managed via the Hugging Face Trainer API, utilizing the AdamW optimizer.

### **Hyperparameter Configuration:**

- Computational Split: 80% train / 20% validation.
- Batch Size: 16.
- Epochs: 3.
- Learning Rate: 0.00005 (HuggingFace default).
- Weight Decay: 0.01.
- Hardware: CUDA-enabled GPU.

## **4.4 Strategy for Class Imbalance**

1. Multiclass Weighting: class weights inversely proportional to class frequencies passed to Cross-Entropy Loss.
2. Binary Reframing: aggregate HS and OL into a single "Harmful" class (83% harmful vs 17% neither).

# **5 Experiments**

## **5.1 Multiclass Model**

We trained the multiclass model to predict one of three labels.

### **Observations:**

- Training accuracy reached  $\sim 0.92$ .
- Validation performance shows slight overfitting.
- Hate speech recall extremely low due to class imbalance.
- Confusion matrix shows  $HS \rightarrow OL$  misclassification dominates.

**Epoch results (multiclass training/validation):**

| Epoch | Training Loss | Validation Loss | Accuracy |
|-------|---------------|-----------------|----------|
| 1     | 0.252000      | 0.266959        | 0.910631 |
| 2     | 0.145900      | 0.260103        | 0.916885 |
| 3     | 0.130600      | 0.294343        | 0.918903 |

## 6 Binary Model

To address class imbalance and ambiguity, labels were collapsed:

- Hate Speech + Offensive Language  $\rightarrow$  Harmful
- Neither  $\rightarrow$  Not Harmful

This reframing aligns with many real-world moderation systems (e.g., Twitter, Reddit) and is much more predictable.

**Binary Performance:**

- Accuracy: 0.9645
- Harmful F1: 0.98
- Not Harmful F1: 0.89

| Binary classification distribution (train set):  | binary_label | count      |
|--|--------------|------------|
|  | harmful      | 20620      |
|  | not harmful  | 4163       |
| Binary classification distribution (percentage): | binary_label | proportion |
|  | harmful      | 0.832022   |
|  | not harmful  | 0.167978   |

## 7 Results / Metrics

Figure 2: Confusion matrix (Multiclass) - schematic shading indicates relative counts (HS / OL / Neither).

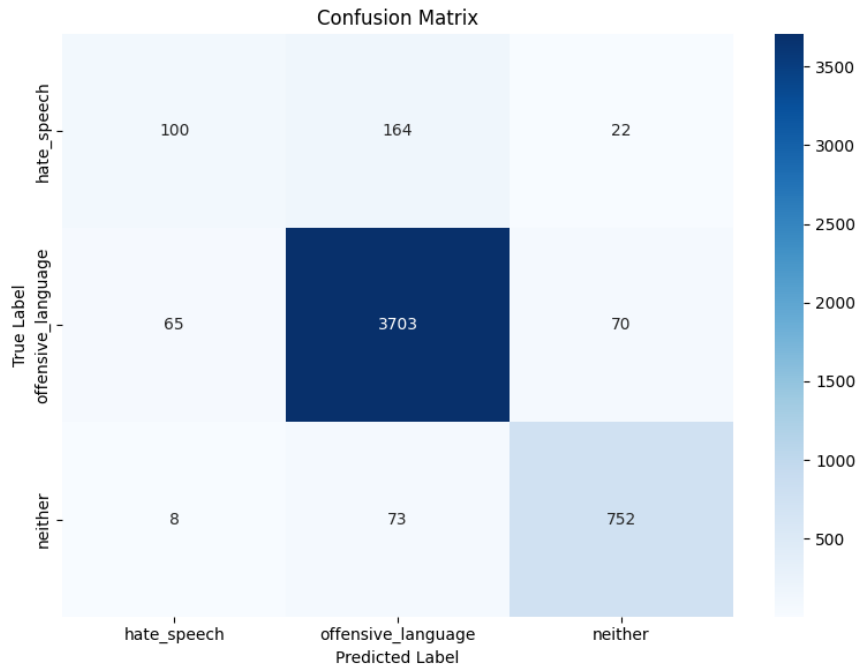


Figure 3: Confusion matrix (Binary) - schematic shading indicates relative counts (Harmful/ Not Harmful).

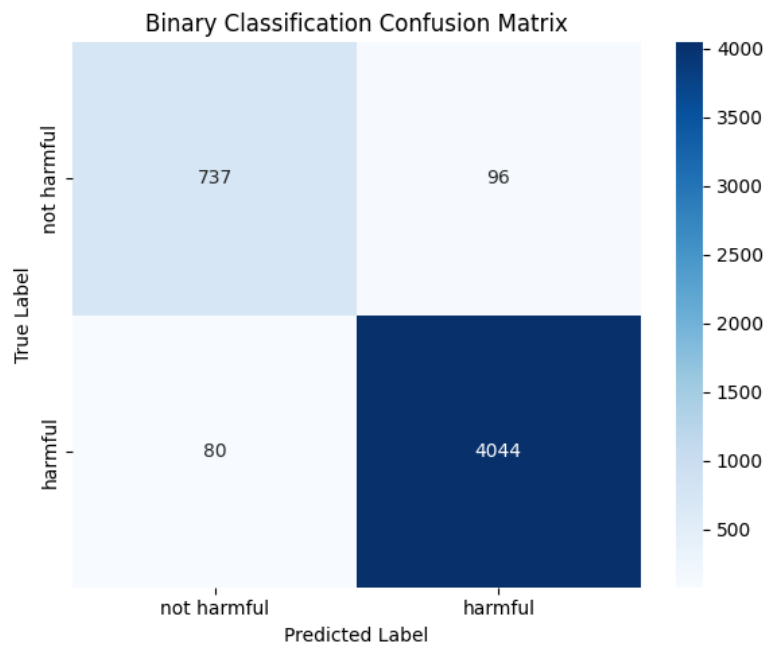


Table 1: Training Dynamics for Multiclass BERT Model (Hate vs. Offensive vs. Neither)

| Epoch    | Training Loss | Validation Loss | Accuracy      |
|----------|---------------|-----------------|---------------|
| 1        | 0.2520        | 0.2670          | 91.06%        |
| 2        | 0.1459        | 0.2601          | 91.69%        |
| <b>3</b> | <b>0.1306</b> | <b>0.2943</b>   | <b>91.89%</b> |

Table 2: Performance Metrics by Class (Multiclass Model)

| Class               | Precision   | Recall      | F1-Score    | Support     |
|---------------------|-------------|-------------|-------------|-------------|
| Hate Speech (0)     | 0.58        | <b>0.35</b> | <b>0.44</b> | 286         |
| Offensive Lang (1)  | 0.94        | 0.96        | 0.95        | 3838        |
| Neither (2)         | 0.89        | 0.90        | 0.90        | 833         |
| <i>Weighted Avg</i> | <i>0.91</i> | <i>0.92</i> | <i>0.91</i> | <i>4957</i> |

Table 3: Training Dynamics for Binary BERT Model (Harmful vs. Non-Harmful)

| Epoch | Training Loss | Validation Loss | Accuracy      |
|-------|---------------|-----------------|---------------|
| 1     | 0.0873        | 0.0933          | 96.57%        |
| 2     | 0.0555        | 0.1191          | <b>96.83%</b> |
| 3     | 0.0442        | 0.1554          | 96.45%        |

Table 4: Performance Metrics by Class (Binary Model)

| Class                   | Precision     | Recall      | F1-Score    | Support     |
|-------------------------|---------------|-------------|-------------|-------------|
| Non-Harmful (0)         | 0.90          | 0.88        | 0.89        | 833         |
| <b>Harmful (1)</b>      | <b>0.98</b>   | <b>0.98</b> | <b>0.98</b> | <b>4124</b> |
| <i>Overall Accuracy</i> | <b>96.45%</b> |             |             |             |

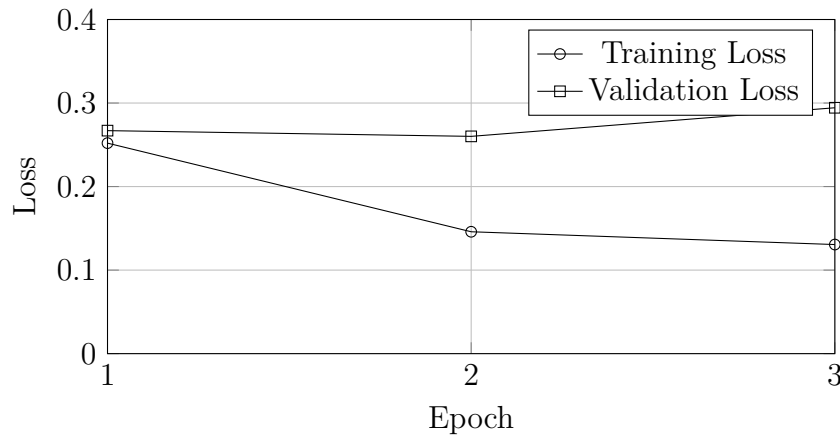


Figure 4: Training and validation loss curves (multiclass).

## 8 Baselines

Baseline Models:

1. Random Classifier: assigns labels uniformly at random. Expected accuracy  $\approx 50\%$  (binary). Expected F1  $\approx 0.5$ .
2. Popularity (Majority-Class) Classifier: predicts harmful for every tweet (since  $\sim 83\%$  harmful). Performance: Accuracy = 83%, F1(harmful)=1.0, F1(neither)=0.

The popularity baseline achieves high accuracy by exploiting class imbalance, but it is functionally useless for moderation because it never identifies safe tweets.

## 9 Error Analysis

### 9.1 False Positives (predicts safe tweets as harmful)

96 cases (examples):

- “i’d never send my kids to private school because then they would never experience ghetto public school fights”
- “Taco Bell is super trash when you’re sober”

Patterns:

- Profanity used in non-hostile contexts
- Quoted lyrics or offensive memes
- Vernacular expressions labeled offensive by dataset
- Friendly insults or referencing harmful language

## 9.2 False Negatives (predicts harmful tweets as safe)

Rare (80 cases), examples:

- “The Steelers new uniforms look like homosexual bumblebees. #wtf?”
- “I said something to you gorilla, stop listening to monkey music”

Reasons:

- Aggression without explicit slurs
- Implicit harmful intent, coded hate speech
- Sarcasm, figurative language
- Attacks on groups without profanity

## 9.3 Clustering of Errors

Observed clusters:

1. Keyword dependency
2. Bias toward majority class
3. Misclassification of subtle implicit harm (length bias)

## 10 Bias Analysis

- Over-reliance on keywords, leads to false positives.
- Class-imbalance leads to false negatives: hate speech is underdetected.
- Slight overfitting (training loss < validation loss).
- Slang can be misclassified because dataset labels vernacular as offensive when in many dialects it is not hateful.

## 11 What we learned

- Multiclass classification is unstable with extreme imbalance.
- BERT performs well but is constrained by noisy training data.
- Binary classification provides a more reliable framing for harmful detection.
- Training data biases propagate through transformer models.
- Simple metrics like accuracy hide imbalanced performance patterns.
- Small improvements scale to millions of users; ethical deployment is required.

## 12 Future work

- Larger, more balanced datasets;
- Domain-specific transformers (e.g., HateBERT);
- Debiasing techniques;
- Thread-level context;
- Expansion to violent threat detection;
- Multilingual/dialect support.

## 13 Conclusion

This project demonstrates that transformer models can effectively identify harmful language, but performance depends heavily on task framing and dataset quality. Our binary BERT classifier achieves strong results and provides a practical foundation for real-world content moderation pipelines. While our project is small, it represents a meaningful contribution toward safer online communication. Our analysis highlights both the potential and limitations of modern NLP systems and the importance of ethical, context-aware evaluation.

## 14 References

- Davidson, T. et al. (2017) ‘Automated Hate Speech Detection and the Problem of Offensive Language’. <http://arxiv.org/abs/1703.04009>
- Devlin, J. et al. (11 Oct 2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” <https://arxiv.org/pdf/1810.0480>
- Fortuna, P. & Nunes, S. (31 Jul 2018). “A Survey on Automatic Detection of Hate Speech in Text.” ACM Computing Surveys (CSUR), Volume 51, Issue 4. Article No.: 85, Pages 1 - 30. <https://doi.org/10.1145/3232676>
- “Potential risks of content, features, and functions: The science of how social media affects youth.” (2024). APA. <https://www.apa.org/topics/social-media-internet/youth-social-media-2024>
- Surdeanu, M. & Valenzuela, M. (2023). Deep Learning for Natural Language Processing: A Gentle Introduction. Cambridge University Press.
- Vaswani, A. et al. (12 Jun 2017). “Attention Is All You Need.” NIPS 2017. <https://arxiv.org/pdf/1706.03762>