Text Classification on Toxic Comments

Ci Song, Shuo Wang
DATASCI 266 - Natural Language Processing with Deep Learning (2024 Spring)

Abstract

Drawing from the detailed investigation into the efficacy of advanced NLP models for toxic comment classification, this paper elucidates the significant advantages of transformer-based models, including BART, BERT, ALBERT, DistilBERT, and RoBERTa, over traditional text vectorization and Naive Bayes classifiers. Through meticulous experimentation on the Jigsaw Toxic Comment Classification dataset, it was revealed that fine-tuned transformer-based models not only substantially improved accuracy, precision, recall, F1 score, and ROC AUC metrics but also demonstrated RoBERTa and DistilBERT's slight superiority across nearly all metrics. This research underscores the transformative impact of leveraging rich contextual and semantic representations for enhancing text classification, offering valuable insights for developing more sophisticated automated moderation tools to combat online toxicity.

I. Introduction

In recent years, Natural Language Processing (NLP) algorithms have undergone remarkable advancements, revolutionizing the field of text analysis and classification. Among these advancements, models like BART (Bidirectional and Auto-Regressive Transformer), BERT (Bidirectional Encoder Representations from Transformers), ALBERT, DistilBERT, and RoBERTa (Robustly Optimized BERT Approach) have demonstrated exceptional performance across various language understanding tasks. In this research project, we focus on harnessing the capabilities of these state-of-the-art algorithms to improve text classification accuracy, particularly in the context of identifying toxic comments.

The task of detecting toxic comments in online discourse presents unique challenges, including the presence of subtle linguistic cues and context-dependent expressions. Traditional machine learning approaches often struggle to capture these nuances effectively. However, the emergence of transformer-based models offers promising solutions by leveraging large-scale pre-training on vast corpora to learn rich representations of language.

Motivated by the potential of these advanced NLP algorithms, we set out to explore their efficacy in enhancing text classification performance on the Jigsaw Toxic Comments dataset. By fine-tuning pre-trained transformer models on this corpus, we aim to leverage their contextual understanding and semantic representations to accurately label comments as toxic or non-toxic.

The significance of this work lies in its potential to contribute to the ongoing evolution of NLP algorithms for real-world applications. By demonstrating the effectiveness of

transformer-based models, including BART, BERT, ALBERT, DistilBERT, and RoBERTa, in toxic comment classification, we not only advance the state-of-the-art in text analysis but also provide practical insights for developers and practitioners seeking to implement automated moderation systems.

In the subsequent sections of this paper, we delve into the methodology employed for model training and evaluation, present our experimental results, and discuss the implications of our findings for the broader NLP research community. Through this focused investigation, we aim to highlight the role of advanced algorithms in driving improvements in text classification accuracy and efficiency.

II. Background

There have been many efforts to classify toxic comments and many research studies were published in recent years. A research study by Khieu and Narwal (Khieu and Narwal, n.d.) from Stanford shows good results on word-level assessment using the LSTM model on both binary and multi-label toxic classifications and they achieved a good accuracy of 87%. Chu and Jue (Chu, Jue, and Wang 2016) later compared a couple of different deep learning algorithms in their study, which achieved 93% accuracy on comment abuse classification problems using word embeddings by RNN with LSTM, CNN and character embeddings by CNN. Another ensemble method proposed by Xie et al (Xie 2022, 429-433) for the multilingual toxic comment classification achieved an AUC of 0.9485 for the validation set.

In this project, we focus on using fine-tuned transformer-based models, including BART, BERT, ALBERT, DistilBERT, and RoBERTa, to compare the classification results and model accuracy for toxic comments classification.

III. Methods

i. Dataset

We used Jigsaw's Toxic Comment Classification Challenge data (Google n.d.) in this project. It contains 159k+ comments that have been human labeled for toxic behavior. The data contains the schema of Id, CommentText, Toxic, ServereToxic, Obscene, Threat, Insult and IndentityHate.

During the data exploration, we found the data was extremely unbalanced. So, we grouped the Labels based on the correlations and decided to have the data with binary labels "Toxic" or "Non-Toxic". We also downsized the training data by randomly selecting 40% of the training data but keeping with the same ratio of the "Toxic" or "Non-Toxic" data. After downsizing the training data, we still have 64K comments in the training dataset.

Also, since we encountered comments in multiple languages (Appendix Figure 1), we incorporated the multilingual versions of BERT and RoBERTa into our models. However, since more than 97% comments are English, the rest models assume the comment is English and ignore the effect of multiple languages.

ii. Models

Baseline Configuration

Our baseline models consist of two traditional text vectorization techniques paired with Naive Bayes classifiers. The CountVectorizer -

Complement Naive Bayes (CNB) and Multinomial Naive Bayes (MNB) models provide benchmarks for word count-based features, while the TfidfVectorizer - CNB and MNB models do the same for TF-IDF weighted features. The Complement Naive Bayes variants are particularly tuned for our imbalanced dataset, using the complement of each class to calculate weights and mitigate majority class bias. Meanwhile, the Multinomial Naive Bayes models apply a more traditional Naive Bayes approach suitable for discrete feature classification. These baselines allow us to establish an initial performance landscape, reflecting the effectiveness of straightforward frequency-based text features against which we will compare the more advanced, context-aware embeddings produced by BART, BERT, ALBERT, DistilBERT, and RoBERTa models.

BART

We adopted the BART architecture (BART n.d.), utilizing the 'facebook/bart-large' checkpoint, for its transformer-based model that pre-trains on a diverse range of text generation tasks and then fine-tunes for specific downstream tasks. Our model capitalizes on BART's sequence classification head for binary classification. It accepts input with a maximum length of 256 tokens and processes it through BART's encoder-decoder architecture. The logits from BART's output are then passed through a flattening layer followed by a dense layer with sigmoid activation to yield the probability of binary class membership. Optimized with a learning rate of 0.00001 using the Adam optimizer, our BART-based model harnesses the sophisticated text generation capabilities of BART, further fine-tuning them for high-stakes classification with commendable accuracy of 92%, reflecting the robustness and versatility of the underlying transformer architecture.

BERT

Our approach exploits the pre-trained capabilities of BERT, adapting its comprehensive language understanding for specific downstream tasks. We propose a direct adaptation model, leveraging BERT's pooled output, a hybrid model that combines BERT with convolutional neural network (CNN) layers to enhance feature extraction, a DistilBERT model, which is a light version of BERT, and an ALBERT model, which reduces memory consumption and increases training speed compared to BERT, making it more scalable.

BERT-Based Classification Model

Our initial model is grounded in the 'bert-base-multilingual-cased' (BERT n.d.) checkpoint, enabling multilingual text processing. It is structured to perform binary classification by directly using BERT's pooled output, which condenses the input sequence's contextual information into a singular comprehensive representation. Following this output, a dense layer with 201 hidden units and ReLU activation is applied to derive higher-order features pertinent to the classification task. To mitigate overfitting, a dropout layer with a rate of 0.3 is integrated, followed by a final dense layer with a sigmoid activation function to compute the probability of the input belonging to one of two classes. The model leverages the Adam optimizer and binary cross-entropy loss for training, focusing on accuracy as the primary performance metric. The network provided us with an accuracy of 91%, which showed that the pre-trained BERT embeddings were effective and showed why it was considered a state-of-the-art language model.

BERT-CNN Hybrid Model for Classification

The second proposition is a hybrid model that combines the rich contextual embeddings

from BERT's last hidden state with the pattern recognition prowess of CNN layers (Appendix Figure 2). This configuration allows the model to capture local contextual clues by extracting features from different-sized n-grams across the text, employing multiple CNN layers with varied kernel sizes. Each CNN layer is followed by global max-pooling to select the most salient features, which are then concatenated and processed through a dropout layer for regularization. The classification decision is rendered through a sigmoid-activated dense layer, mirroring the optimization and loss computation strategies of the first model. This hybrid approach aims to enhance classification performance by leveraging both global and local textual features. This model showed 89% accuracy, which slightly improved from the last BERT-based model.

DistilBERT-Based Classification Model

Following the success of the BERT and BERT-CNN hybrid models, we explored the efficiency of DistilBERT (DistilBERT n.d.), a lighter version of BERT that retains most of its predecessor's performance while being more resource-efficient. Our DistilBERT-based model is built upon the 'distilbert-base-uncased' checkpoint, specifically designed for English language processing. The model architecture adopts DistilBERT's transformer layers, condensing the textual context into a sequence of embeddings. Similar to the BERT-based model, the classification head starts with the first token's embedding—assumed to encapsulate the sequence's overall context—followed by a dense layer with 275 hidden units and ReLU activation to distill relevant features for the classification task. A dropout layer with a rate of 0.3 is employed to enhance the model's generalization capabilities. The final output layer uses a sigmoid activation function to yield the probability of the input text being classified into one of the binary categories. Optimized with Adam, binary cross-entropy and a learning rate of 0.00001, this model emphasizes accuracy while benefiting from the distilled knowledge of BERT, achieving significant computational savings. In our experiments, the DistilBERT-based model demonstrated an impressive accuracy of 92%, validating the effectiveness of knowledge distillation in transformer-based architectures for the task of binary text classification.

ALBERT-Based Classification Model

In pursuit of refining the balance between model complexity and performance, we integrated the ALBERT model (ALBERT n.d.) into our study. ALBERT, or "A Lite BERT," further optimizes BERT for performance and size by factorizing the embedding layer and sharing parameters across layers. Our model utilizes the 'albert-base-v2' checkpoint, which provides a solid foundation for understanding and processing language with reduced parameters without a substantial compromise on contextual comprehension. Our ALBERT-based classification model employs the same structural principles as the previous models, commencing with ALBERT's pooler output to secure a dense representation of the input sequence. The subsequent architecture encompasses a dense layer with 201 hidden units paired with ReLU activation, crafted to unravel complex features relevant to the binary classification objective. Following the dense layer, we incorporate a dropout layer with a 0.3 rate to ensure robustness against overfitting. The binary classification output is then derived through a sigmoid-activated dense layer. The model is compiled with the Adam optimizer, utilizing binary cross-entropy loss, and centers accuracy as the pivotal evaluation metric. Our ALBERT classification model was able to reach an accuracy of 89%.

This achievement underscores the efficacy of ALBERT's streamlined architecture in maintaining high levels of performance, affirming its suitability for tasks requiring both linguistic nuance and computational efficiency.

RoBERTa

We then developed a binary classification model employing the XLM-RoBERTa (jplu/tf-xlm-roberta-base) (XLM-RoBERTa n.d.) architecture, leveraging its proficiency in understanding multiple languages for a language-agnostic text classification task. XLM-RoBERTa, an evolution of BERT architecture, brings several enhancements that make it particularly suited for our purposes. Unlike BERT, which is pre-trained using a combination of masked language modeling and next sentence prediction, RoBERTa relies solely on masked language modeling but with optimized training strategies, such as dynamic masking, larger mini-batches, and more data. These improvements enable RoBERTa to understand context and semantics with greater accuracy, making it an ideal choice for complex natural language understanding tasks like ours.

Our model processes tokenized text using input_ids and attention_mask, handling sequences up to a set maximum length. At its heart lies the XLM-RoBERTa transformer, where we derive a comprehensive representation from the hidden state of the initial "CLS" token for classification tasks. To better capture complex patterns in binary classification, we've added a dense layer with 275 units and ReLU activation. The final output layer features a single unit with sigmoid activation to determine the probability of an input belonging to the positive class. The model uses the Adam optimizer, is compiled with binary cross-entropy loss, a learning rate of 0.00001, and focuses on accuracy as the key performance indicator.

This approach enhances the model's ability to generalize from pre-learned linguistic patterns to the task-specific nuances of our binary classification challenge. Lastly, the RoBERTa model achieved accuracy of 93%.

IV. Experimentation Results and Discussion

In this project, we implemented various models including CountVectorizer with Complement Naive Bayes (CNB) and Multinomial Naive Bayes (MNB), TfidfVectorizer with CNB and MNB, as well as advanced deep learning models like BART, BERT, BERT with Convolutional Neural Network (BERT+CNN), distilBERT, ALBERT, and RoBERTa. The performance metrics considered are Accuracy, Precision, Recall, F1 Score, and ROC AUC (Ralf Krestel Betty van Aken 2018).

Table 1(Appendix) is the result of our models. Models with Transformers (BART, BERT, and RoBERTa etc.) beat the model performances for all evaluation metrics compared to the baseline model with Naive Bayes classifiers.

The CountVectorizer - CNB model shows moderate performance with an accuracy of 89%, precision of 92%, recall of 89%, an F1 score of 90%, and a ROC AUC of 81.26%. The CountVectorizer - MNB model improves in terms of accuracy (91%), recall (91%) and F1 Score (92%), keeps the same precision, and has a lower ROC AUC (78.30%).

The TfidfVectorizer models outperform the CountVectorizer models in terms of Accuracy and Recall, with the CNB variant achieving an

accuracy of 91% and a recall of 91% and the MNB variant achieving an accuracy of 92% and a recall of 92%. The TfidfVectorizer models underperform the CountVectorizer models in terms of Precision and ROC AUC, with CNB variant achieving a precision of 90% and a ROC AUC of 70.22% and the MNB variant achieving a precision of 91% and a ROC AUC of 57.63%. There is a significant decrease for the ROC AUC for TfidfVectorizer models.

The deep learning models, BART, BERT, BERT+CNN, distilBERT, ALBERT, and RoBERTa, significantly outperform the previous models for ROC AUC with in a range from 96.64% to 97.04%.

BERT achieves an accuracy of 91%, a precision of 94%, a recall of 91%, an F1 score of 92%, and a ROC AUC of 96.22%. BERT+CNN shows a slight improvement in accuracy (96.57%), indicating the addition of CNN layers might slightly enhance model performance.

DistilBERT has the lowest running time (1.77 h for 2 epochs with T4 GPU), and BART has the highest running time (2.86 h for 2 epochs with V100 High-RAM) among deep learning models.

RoBERTa performs best overall among all deep learning models, achieving the highest accuracy (93%), precision (94%), recall (93%), and an F1 score of 93%, also has comparatively high ROC AUC of 96.64%, suggesting it is the most effective model among those evaluated for this specific task.

Future Work

In future work, we identify two primary areas for potential exploration to enhance or validate our findings. Initially, an analysis of the raw data revealed significant imbalances in two scales, first, the “Non-Toxic” / “Toxic" labels ratio is about 9:1. Second, the “Toxic” labels are subdivided into multiple labels including Toxic, ServereToxic, Obscene, Threat, Insult and IndentityHate. The labels, which consist of the combination of one or more sub labels, are also significantly imbalanced across the various categories labeled as 'Toxic.' This imbalance led to a lack of true positive predictions in our multi-class text classification, prompting a shift towards binary classification. A potential solution to deal with imbalance data for multi-class classification is to build a two-step models: (1) a binary classification model for overall data to predict the “Toxic” and the “Non-Toxic” labels, and (2) a multiclass classification model for predicted “Toxic” labels to further predict the sub labels under “Toxic” labels.

Furthermore, we experimented with advanced NLP algorithms, including RoBERTa-LONG, T5, and XLNET. However, the limited GPU capacity available through Google Colab resulted in Resource Exhausted Errors. To circumvent these challenges, we are considering the adoption of parallel computing techniques or securing additional resources. This would enable a more thorough evaluation of these algorithms' effectiveness in improving model performance.

V. Conclusion

In conclusion, the comprehensive analysis conducted in this study underscores the superior performance of the advanced deep learning models (including BART, BERT, distilBERT, ALBERT, and RoBERTa) over traditional text vectorization and naive Bayes classifiers for the task of toxic comment binary classification. These fine-tuned transformer-

based models achieved exemplary scores for ROC AUC while keeping similar scores across accuracy, precision, recall, and F1 score. Notably, RoBERTa marginally outperformed among deep learning models in nearly every assessed metric, establishing itself as the premier model for this specific application. This evidence strongly supports the conclusion that transformer-based NLP technologies, with their deep understanding of context and semantics, significantly elevate the capabilities of text classification systems. Such advancements hold significant promises for the development of more effective automated moderation tools, offering practical solutions to manage and mitigate online toxicity.

References

BERT. n.d. "BERT multilingual base model (cased)." Hugging Face. https://huggingface.co/google-bert/bert-base-multilingual-cased.

Betty van Aken, Julian Risch, Ralf Krestel, Alexander Löser. 2018. "Challenges for Toxic Comment Classification: An In-Depth Error Analysis." 33-42.

Chu, Theodora, Kylie Jue, and Max Wang. 2016. Comment abuse classification with deep learning. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762092.pdf.

Google, Jigsaw and. n.d. ""Toxic Comment Classification Challenge."." Kaggle. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

Khieu, Kevin, and Neha Narwal. n.d. Detecting and Classifying Toxic Comments. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf.

Xie, Gaofei. 2022. An ensemble multilingual model for toxic comment classification. Vol. 12176. International Conference on Algorithms Microchips and Network Applications.

XLM-RoBERTa. n.d. "XLM-RoBERTa." Hugging Face. https://huggingface.co/docs/transformers/model_doc/xlm-roberta.

BART. n.d. "BART Base Model." Hugging Face. https://huggingface.co/facebook/bart-base.

ALBERT. n.d. "ALBERT Base v2 Model." Hugging Face. https://huggingface.co/albert-base-v2.

DistilBERT. n.d. "DistilBERT Base Uncased Model." Hugging Face. https://huggingface.co/distilbert-base-uncased.

Appendix

Table 1: Model Evaluation Metrics Summary

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Runing Time	Hardware
CountVectorizer - CNB	0.89	0.92	0.89	0.90	0.8126	<10s	CPU
CountVectorizer - MNB	0.91	0.92	0.91	0.92	0.7830	<10s	CPU
TfidfVectorizer - CNB	0.91	0.90	0.91	0.91	0.7022	<10s	CPU
TfidfVectorizer - MNB	0.92	0.91	0.92	0.89	0.5763	<10s	CPU
DAN-Static	0.91	0.89	0.91	0.89	0.8498	< 2min (10 epochs)	CPU
DAN-Retrain_word2vec	0.91	0.91	0.91	0.91	0.8711	< 10min (10 epochs)	CPU
DAN-Retrain_uniform	0.91	0.90	0.91	0.91	0.8713	< 10min (10 epochs)	CPU
WAN	0.91	0.91	0.91	0.91	0.8782	< 10min (10 epochs)	CPU
CNN-non_Retrain	0.90	0.91	0.90	0.90	0.8821	< 5min (10 epochs)	CPU
CNN-Retrain	0.90	0.90	0.90	0.90	0.8515	< 10min (10 epochs)	CPU
RNN-non_Retrain	0.90	0.91	0.90	0.91	0.8839	< 5min (10 epochs)	CPU
RNN-Retrain	0.89	0.90	0.89	0.90	0.8445	< 10min (10 epochs)	CPU
CNN+RNN-non_Retrain	0.89	0.90	0.89	0.89	0.8662	< 5min (10 epochs)	CPU
CNN+RNN-Retrain	0.90	0.90	0.90	0.90	0.8543	< 10min (10 epochs)	CPU
Bidirectional_GRU	0.90	0.93	0.9	0.91	0.9287	< 1h (10 epochs)	T4 GPU
BART	0.92	**0.94**	0.92	**0.93**	0.9666	2.86 h (2 epochs)	V100-High RAM
ALBERT-TFAlbertModel	0.89	**0.94**	0.89	0.90	0.9689	3h (2 epochs)	T4 GPU
ALBERT-TFAlbertForSequenceClassification	0.90	0.82	0.90	0.86	0.5061	3h (2 epochs)	T4 GPU
DistilBERT	0.92	**0.94**	0.92	**0.93**	**0.9704**	1.77h (2 epochs)	T4 GPU

DistilBERT+CNN	0.89	**0.94**	0.89	0.90	0.9701	1.88h (2 epochs)	T4 GPU
BERT	0.91	**0.94**	0.91	0.92	0.9622	3.67h (2 epochs)	T4 GPU
BERT+RNN	0.89	**0.94**	0.89	0.90	0.9618	3.91h (2 epochs)	T4 GPU
BERT+CNN	0.89	**0.94**	0.89	0.90	0.9657	3.8h (2 epochs)	T4 GPU
RoBERTa	**0.93**	**0.94**	**0.93**	**0.93**	0.9664	3.72h (2 epochs)	T4 GPU

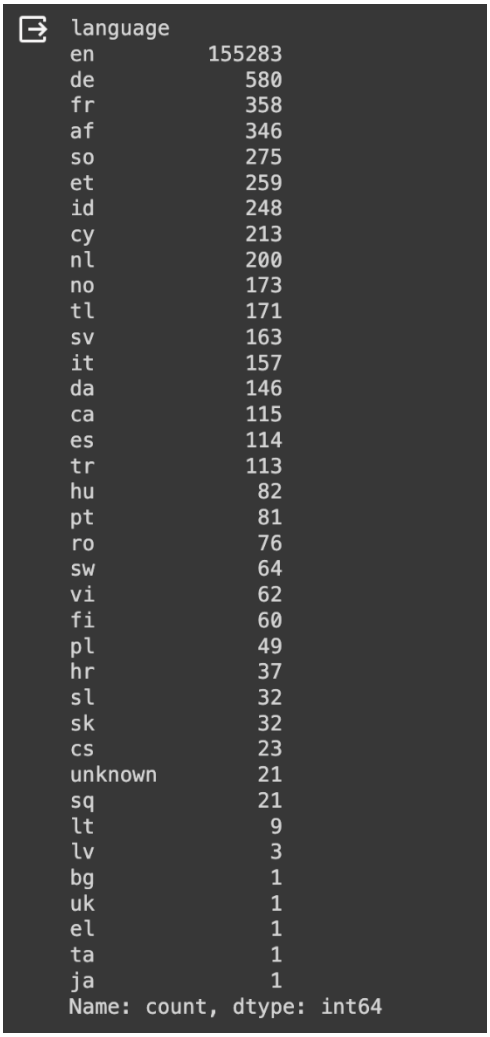

Figure 1: Comment Language Counts

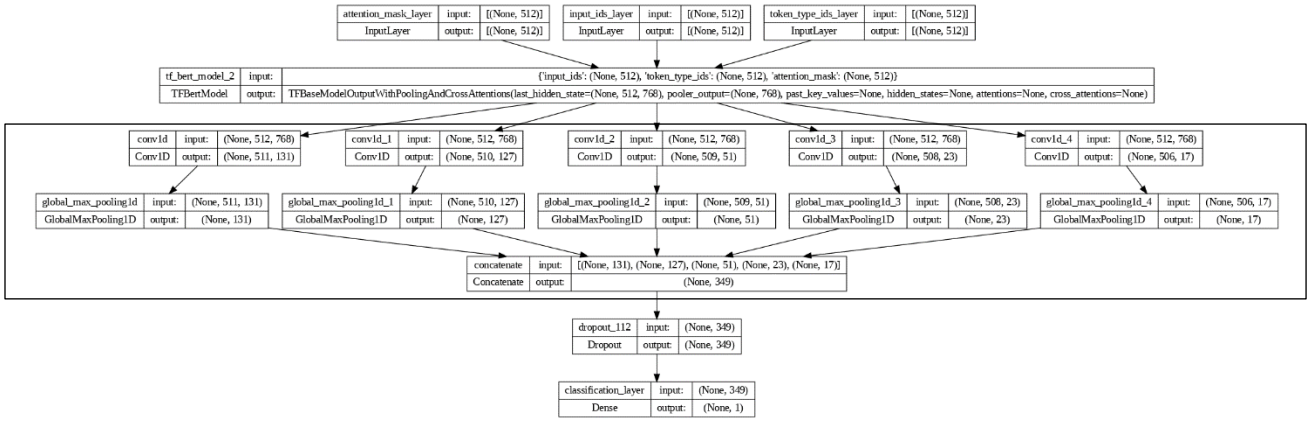

Figure 2: BERT+CNN

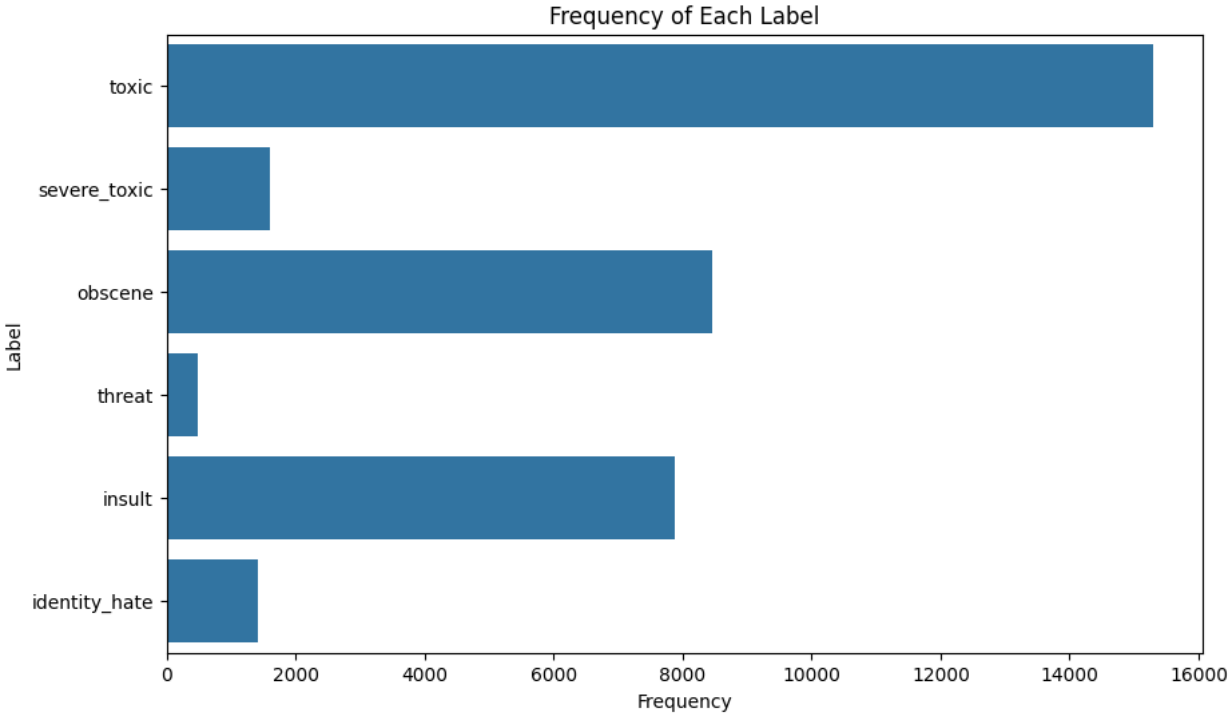

Figure 3: Label Frequency

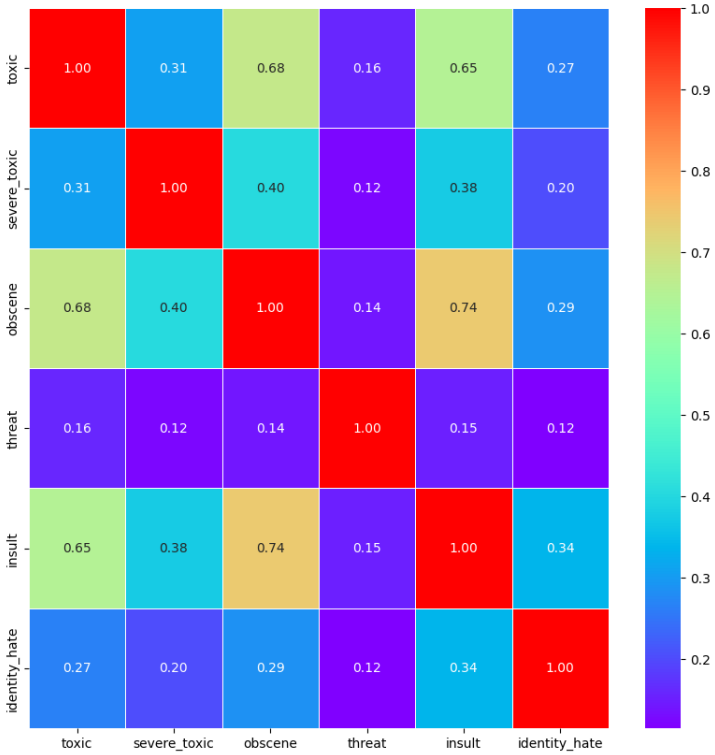

Figure 4: Label Correlations