Detecting Sarcasm, Toxic Comments, and Hateful Speech in Text

Ayush Dhoundiyal

Abstract

In the era of online communication, detecting sarcasm, toxic comments, and hateful speech has become crucial for improving user experiences and moderating harmful content. Despite the distinct nature of these tasks, their shared dependence on understanding nuanced context-dependent language enables overlapping methodologies for efficient detection. This research focuses on developing three specialized systems for these tasks, using pre-processing techniques, feature extraction, and machine learning models. In this paper we were able to find fine-tune models, work with simpler models to get required classification results for each of the three tasks. Logistic Regression was one of the consistent performing models across all three for the classification task along with bert-based fine-tuned model.

1 Introduction

The goal is to build three best separate models, each focused on one task: sarcasm detection, toxic comment detection, and hateful speech detection. The challenge lies in ensuring that each system accurately identifies its respective target in text while leveraging similar model architectures and data processing techniques to streamline implementation.

These tasks are essential for platforms aiming to improve content moderation, sentiment analysis, and user interactions. Automating the detection of sarcasm, toxicity, and hate speech helps prevent miscommunication, reduce harmful interactions, and enhance content recommendations. Addressing these challenges enables platforms to foster safer and more engaging online environments.

Although each task is distinct, they share the underlying challenge of understanding nuanced language. This overlap makes it feasible to approach them using similar models, enabling a unified framework to tackle these complex problems. Additionally, deeper insights into these tasks can highlight their real-world impact and guide the development of more effective and scalable solutions.

2 Related Work

Sarcasm Detection: There has been a lot of ongoing and completed research done for sarcasm detection where people are trying to classify online comments as sarcastic or not. But even with a lot of research, the model gets the average accuracy in the online text like (Šandor & Babac, 2023). In this paper, our objective is to improve the accuracy of the model, identify shortcomings, and analyze the results. Furthermore, we examine the impact of datasets and context on sarcasm detection by comparing models with simpler datasets, providing a basis for our findings.

Hate Speech Detection: There has been significant work in the field of hate speech detection. For example, The (Mathew et al., 2021) HateXplain paper tackles hate speech classification using a dataset similar to ours, employing a multi-class approach with categories for hate speech, offensive language, and neither. Drawing inspiration from their methodology, we adapt the dataset for binary classification: merging offensive language with "none" in one configuration and combining offensive and hate speech in another.

While the above stated study leverages a variety of models, with a primary focus on their own model and embeddings, their work also demonstrates strong performance with BERT-based models. Based on this, we explore BERT and XGBoost, leveraging their high performance for binary classification tasks for imbalance in dataset.

Toxic Comment Detection: There has been significant work in the field of Toxic Comment Detection. Here, we aim to build upon the work of (Zaheri et al., 2020). In the paper, they confirm our approach of combining different class labels to form toxic and non-toxic comment sections. This approach makes more sense when we consider that the dataset they used to draw their inferences is very similar to ours.

3 Methodology

Sarcasm Detection: We trained and evaluated multiple models, including Logistic Regression, Naive Bayes, MLP Classifiers, BERT Base, and DistilBERT. Logistic Regression was tested using TF-IDF and Word2Vec embeddings, while GloVe Twitter-25 embeddings were selected for their similarity to Twitter-like comment structures. Upon initial result to broaden our analysis, DistilBERT was fine-tuned on the nikesh66/Sarcasm-dataset.
Hate Speech Detection System: We experimented with multiple models like Logistic Regression (LR) with TF-IDF and GloVe Twitter 200 embeddings, both with and without SMOTE oversampling. XGBoost with TF-IDF embeddings, Multi-Layer Perceptron (MLP) with GloVe Twitter 200 and Word2Vec Google News embeddings (200D and 300D) and Facebook's 'roberta-hate-speech-dynabench-r4-target' model from Hugging Face.

To combat class imbalance, SMOTE oversampling technique which augments the data to create virtual data for imbalance class in data. We also combined different set of labels and also fine tuned the distill-bert-uncased to get the state of the art result and create a vast analysis for the task.
Toxic Comment Detection: Models either trained from scratch or fine tuned for the task included Logistic Regression, Naive bayes and fine-tuned 'bert-base-uncased'. We also evaluated our dataset on 'martin-ha/toxic-comment-model' from Hugging Face for get our fine tuned model comparison.

Metrics such as accuracy, precision, recall, and F1 score were used across all three tasks to evaluate model performance, helping identify overall accuracy and class imbalances.

4 Experiments

4.1 Sarcasm Detection

The dataset consisted of approximately 1,000,000 Reddit entries obtained from Kaggle[4], with 9 descriptive columns: label, comment, author, subreddit, score, ups, downs, date, createdutc, and parentcomment. Only the 'comment' was used as the input feature, while 'label' served as the target variable. The data was evenly distributed, so no sampling was required, and the baseline was set at 50%.

Since there was no predefined train-test split, a 70/30 split was applied, with 70% used for training and 30% for testing. Preprocessing steps included converting all comments to lowercase, removing punctuation, stopwords, and lemmatizing words. These preprocessing steps were consistent with those applied in previous datasets.

We began by evaluating Logistic Regression with TF-IDF embeddings, testing with default hyperparameters while tuning the n-gram range of the TF-IDF feature extraction. We compared two configurations: one without any n-grams and the other with a combination of unigrams and bigrams (1,2 n-grams). The latter configuration yielded better results.

We also trained a Naïve Bayes classifier using the same TF-IDF hyperparameters. Additionally, we explored Logistic Regression models using GloVe embeddings for feature extraction and compared their performance with other embeddings.

Furthermore, we extended our experiments to Multi-Layer Perceptron (MLP) models to assess their learning capabilities and flexibility on this task.

We additionally trained two pre-trained transformer models, namely BERT-base-uncased and DistilBERT-uncased, using 70% of the data for training and the remaining 30% for evaluation.

To further investigate, we conducted additional experiments on a simpler dataset, nikesh66/Sarcasm-dataset , available on Hugging Face. This dataset was context-independent, on which we fine-tune the DistilBERT model. These experiments yielded outstanding results with accuracy of **0.94**, precision of **1.0**, recall of **0.99** and F1 score of **0.94**.

4.2 Hate Speech Detection

The dataset for hate speech detection contained fewer than 2.5k entries for hate speech, with fields such as 'id', 'count', 'hate_speech', 'offensive_language', 'neither', 'class', and 'tweet'. The 'class' column had three values (0 for hate speech, 1 for offensive language, 2 for neither) based on majority labeling. To focus on binary detection (Hate Speech vs. Non-Hate Speech), we combined 'offensive_language' and 'neither' into one label (0), while retaining 'hate_speech' as the other label (1).

After visualization, the dataset showed a significant class imbalance: 2k instances of hate speech vs. 23k non-hate speech instances. We split the data into 80% training and 20% testing, using stratified sampling to maintain class proportions. This approach ensured that the same proportion of class labels was represented in both training and testing datasets, thereby avoiding data imbalance issues during splitting.

We first experimented with Logistic Regression (LR) using two embeddings like TF-IDF with n-grams: We tested both unigram and bigram n-grams and GloVe Twitter embeddings.

SMOTE Technique with different models was used along with TF-Idf and Glove-Twitter-Embeddings.

We also trained XGBoost and compared its performance to other baseline models.

After achieving average result with initial combination of class label we combined offensive and HateSpeech as class label 1 and None as class label 0. This change allowed us to combine these two categories and achieve remarkable performance improvements in all the models.

Subsequently, we fine-tuned the facebook/roberta-hate-speech-dynabench-r4-target model using our custom preprocessing pipeline and retrained it with an 80/20 train-test split.

We also fine-tuned the bert-base-uncased model on the same 80/20 split dataset and obtained exceptional results across all metrics after applying these adjustments with accuracy of **0.96**, precision of **0.97**, recall of **0.97** and F1 score of **0.97**.

4.3 Toxic Comment Detection

The dataset included fields such as ‘id‘, ‘comment_text‘, ‘toxic‘, ‘severe_toxic‘, ‘obscene‘, ‘threat‘, ‘insult‘, and ‘identity_hate‘, with all but ‘comment_text‘ as integers. It contained 160k entries with no missing values. The target variable was set to 1 for any toxic category (‘toxic‘, ‘severe_toxic‘, ‘obscene‘, ‘threat‘, ‘insult‘, ‘identity_hate‘) and 0 otherwise.

The ‘toxic‘ category had the most instances (14,000+), followed by ‘obscene‘ and ‘insult‘ (8,000+ each). In contrast, ‘severe_toxic‘ and ‘identity_hate‘ had fewer than 2,000 entries, and ‘threat‘ had fewer than 200. A 70/30 train-test split was applied, with balanced splitting.

For toxic comment detection, we optimized the Logistic Regression model by tuning the regularization parameter C. We experimented with various values of C. All other parameters were kept as default. Logistic Regression was combined with TF-IDF embeddings, yielding satisfactory results.

In addition to Logistic Regression, we also combined Naïve Bayes with TF-IDF embeddings, achieving excellent accuracy and a precision score of **1.0**. We further fine-tuned the ‘bert-base-uncased‘ model by running it till 6 epoches and applying early stopping with a train-test split of 70/30. This fine-tuned model demonstrated significantly better accuracy of **0.96**, precision of **0.95** and F1 score of **0.71**.

Furthermore, we compared our fine-tuned model with the open-source ‘martin-ha/toxic-comment-model‘ hosted on Hugging Face by testing our our dataset.

5 Results and Analysis

5.1 Sarcasm Detection

Table 1: Performance of Models for Sarcasm Detection

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression (tf-idf no n-gram)	0.67	**0.69**	0.64	0.66
Naïve Bayes (tf-idf no n-gram)	0.65	0.64	0.68	0.66
Logistic Regression (tf-idf 1,2 n-gram)	**0.68**	**0.69**	0.67	**0.68**
Naïve Bayes (tf-idf 1,2 n-gram)	0.66	0.65	**0.70**	**0.68**
Logistic Regression (GloVe)	0.56	**0.69**	0.21	0.32
MLP (GloVe)	0.65	0.66	0.61	0.64
bert-base-uncased	**0.68**	**0.67**	0.50	0.57
distill-bert-uncased	0.66	0.60	**0.57**	**0.58**

For sarcasm detection, we initially experimented with Logistic Regression and observed better performance when combining it with TF-IDF vectorization using both unigram and bigram (1,2) n-grams. This improvement can be attributed to the model's ability to generalize beyond individual words by incorporating contextual patterns captured by bigrams. However, the overall performance remained suboptimal, with the best results yielding an accuracy of **0.68**, a precision of **0.69**, and an F1 score of **0.68** using Logistic Regression with TF-IDF and n-gram hyperparameters set to 1,2(Can be seen in Table 1).

We next tested Naïve Bayes with TF-IDF embeddings, exploring configurations with no n-grams and 1,2 n-grams. While there were slight improvements, the results were still unsatisfactory. To address these limitations, we incorporated GloVe Twitter-25 embeddings with both Logistic Regression and an MLP classifier, hypothesizing that the similarity between tweet-style data and Reddit comments might enhance performance. While these models performed better than the baseline, their overall results were average and failed to achieve satisfactory generalization(Refer Table 1).

We inferred that simpler models struggled due to insufficient data or complexity in the task, which limited their ability to generalize effectively. Consequently, we explored fine-tuned transformer-based models, as detailed in Table 1. However, these models also struggled to generalize, likely because sarcasm detection is inherently challenging, especially when contextual information from parent threads—which often provides critical cues for sarcasm—was not included in the dataset.

Table 2: Results DistilBERT (Fine-Tuned) on nikesh66/Sarcasm-dataset

Model	Accuracy	Precision	Recall	F1 Score
distill-bert-uncased	0.94	1.0	**0.89**	**0.94**

To validate our hypothesis, we fine-tuned a DistilBERT model using the Hugging Face dataset 'nikesh66/Sarcasm-dataset', which contains approximately 5,000 tweet samples. This dataset, being smaller and more focused, better captured sarcasm. The results confirmed our assumptions, with the fine-tuned model achieving state-of-the-art performance: an accuracy of **0.94**, precision of **1.0**, and recall of **0.89**, as detailed in Table 2. These findings emphasize the importance of using datasets with richer contextual information and task-specific characteristics for sarcasm detection.

5.2 Hate Speech Detection System

Hate speech classification involves distinguishing between hate speech and non-hate speech. Initially, the dataset was modified by combining the offensive language and none categories into a single label (0), while assigning the hate speech class the label (1). Using this modified dataset, we began by testing simpler models like Logistic Regression and Naïve Bayes.

For Logistic Regression, we experimented with both TF-IDF embeddings and GloVe Twitter-200 embeddings. For Naïve Bayes, we used TF-IDF embeddings. Although these models achieved decent accuracy, their precision and recall scores were poor, as they failed to classify both classes evenly (reference Table 3). This highlighted the model's inability to handle the class imbalance in the dataset.

To address this, we employed the SMOTE oversampling technique, which improved performance slightly, but the improvements were not significant across all metrics. Subsequently, we incorporated the XGBoost model, which is well-known for its robustness in handling imbalanced datasets. XGBoost achieved the highest accuracy among the simpler models; however, the recall remained low, and precision, while improved, only reached 0.51, which was still insufficient.

Table 3: Initial Results (Hate Speech Detection)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.91	0.34	**0.62**	0.44
Naïve Bayes (no n-gram)	**0.94**	0.00	0.00	0.00
LR with TF-IDF (with oversampling)	0.92	0.38	0.60	0.47
LR with GloVe Twitter 200 (no oversampling)	**0.94**	0.42	0.21	0.28
LR with GloVe Twitter 200 (with oversampling)	**0.94**	0.44	0.12	0.19
XGBoost with TF-IDF	**0.94**	**0.51**	0.26	0.35
MLP with GloVe Twitter 200 (no oversampling)	**0.94**	0.42	0.22	0.29
MLP with GloVe Twitter 200 (with oversampling)	0.92	0.42	0.33	0.33
MLP with Word2Vec (oversampling)	0.90	0.24	0.37	0.29
MLP with Word2Vec (no oversampling)	0.93	0.37	0.21	0.27

Next, we tested an MLP classifier using Word2Vec (Google News 300) and GloVe Twitter-200 embeddings. While the accuracy was among the highest achieved, precision and recall remained subpar. To further analyze the dataset, we evaluated an open-source Hugging Face model (facebook/roberta-hate-speech-dynabench-r4-target) and observed an accuracy of only 0.24, demonstrating the model's inability to classify effectively under the current configuration.

Upon deeper analysis, we identified significant overlap between hate speech and offensive language categories, which likely contributed to poor classification results. To address this, we merged offensive language and hate speech into a single class (1) and retained none as class (0). This adjustment validated our hypothesis, with models showing remarkable improvements across all metrics (see Table 4). Logistic Regression emerged as one of the top-performing simpler models with **0.94**, precision of **0.99**, and recall of **0.94**, closely followed by XGBoost.

We also evaluated the Facebook model with and without preprocessing. Results showed better performance with our preprocessing pipeline. Building on this, we fine-tuned the Facebook model using the preprocessed data and observed significant improvements. Lastly, we fine-tuned our own pre-trained BERT-base-uncased model, achieving state-

Table 4: New Results (Hate Speech Detection) after combining hate speech and offensive language as 1 label

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.94	**0.99**	0.94	0.96
Naïve Bayes (tf-idf no n-gram)	0.84	0.84	**1.0**	0.91
LR with TF-IDF (with oversampling)	0.94	0.98	0.95	0.96
LR with GloVe Twitter 200 (no oversampling)	0.63	0.99	0.56	0.72
LR with GloVe Twitter 200 (with oversampling)	0.67	0.99	0.60	0.75
XGBoost with TF-IDF	0.95	0.98	0.95	**0.97**
MLP with GloVe Twitter 200 (no oversampling)	0.92	0.95	0.96	0.95
MLP with GloVe Twitter 200 (with oversampling)	0.92	0.95	0.95	0.95
MLP with Word2Vec Google News (oversampling)	0.92	0.96	0.94	0.95
MLP with Word2Vec Google News (no oversampling)	0.92	0.95	0.95	0.95
facebook/roberta-hate-speech-dynabench-r4-target (without preprocessing)	0.93	0.93	0.93	0.93
facebook/roberta-hate-speech-dynabench-r4-target (with preprocessing)	0.94	0.95	0.94	0.95
facebook/roberta-hate-speech-dynabench-r4-target (fine-tuned)	0.95	0.98	0.96	**0.97**
bert-base-uncased	**0.96**	0.97	0.97	**0.97**

of-the-art metrics across all evaluation criteria. The significant improvement in results with **0.96**, precision of **0.97**, recall of **0.97**, and F1 score **0.97** as seen in (Table 4). This highlights the importance of addressing overlap between offensive language and hate speech categories. Further analysis on the relationship between these categories is warranted to refine classification performance.

5.3 Toxic Comment Detection

The dataset for toxic comment detection was substantial, allowing us to explore multiple classification approaches. We began with simpler models, which performed well, as shown in Table 5. Significant hyperparameter tuning was conducted for Logistic Regression since the model demonstrated excellent recall, which was our primary objective—achieving a model with both high accuracy **0.94** and outstanding recall **0.83**.

Table 5: Performance of Models for Toxic Comment Detection

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression (C=2)	0.94	0.68	**0.83**	**0.75**
Naïve Bayes	0.91	**1.00**	0.09	0.17
bert-base-uncased	**0.95**	0.84	0.62	0.71
martin-ha/toxic-comment-model	0.94	0.85	0.51	0.64

We focused on tuning the hyperparameter C (the inverse of regularization strength) by testing values such as 0.1, 1, 2, 3, and 5. The model performed best with $C = 2$, delivering respectable metrics, including high accuracy while maintaining a decent recall score.

Although the results were satisfactory with Logistic Regression, we extended our experiments to evaluate the performance of a fine-tuned DistilBERT model. The fine-tuned model achieved exceptional accuracy of 0.95 and impressive precision, but the recall was comparatively lower. Additionally, we tested an open-source Hugging Face model (‘martin-ha/toxic-comment-model‘) on our dataset. Our fine-tuned models outperformed the open-source model across all key metrics, demonstrating their superior performance.

6 Conclusion

The models performed well across sarcasm detection, hate speech classification, and toxic comment detection, despite variations due to dataset characteristics. For **sarcasm detection**, models excelled on simple tweet datasets but struggled with complex datasets requiring contextual understanding, highlighting the need to incorporate context during training. Meanwhile, for **hate speech detection**, combining "hate speech" and "offensive" as one label and treating "none" as another yielded strong results. Future work should focus on separating hate speech from offensive language.

Additionally, **toxic comment classification** exhibited high accuracy overall. Logistic Regression achieved the best recall, which is a crucial metric for this task. Further exploration of pre-trained models is needed to enhance recall.

Overall, Logistic Regression and the fine-tuned BERT-base-uncased model consistently delivered strong performance across all three tasks, making them reliable choices for classification problems. This highlights their potential for further development and deployment in real-world applications. (Click on this link to be redirected to the GitHub page for code implementation)

References

Šandor, D., & Babac, M. B. (2023). Sarcasm detection in online comments using machine learning. *Information Discovery and Delivery*, (ahead-of-print).

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI conference on artificial intelligence*, *35*(17), 14867–14875.

Zaheri, S., Leath, J., & Stroud, D. (2020). Toxic comment classification. *SMU Data Science Review*, *3*(1), 13.