Hate Speech Analysis

Younes Hourri, Mackenzie Stathis, Rakesh Khanna
260969342, 260987888, 260944862
Github Link : https://github.com/kesh-khanna/hate_analysis

Abstract

Hate speech detection remains a significant challenge on social media due to its inherent ambiguity and context dependency. In this paper, we compare various classical machine learning and deep learning methods in differentiating between hate speech and non-hateful discourse, utilizing the UC Berkeley hate speech dataset [Kennedy et al., 2020]. While the deep learning models were fine-tuned using pre-trained embeddings, the classical methods employed a TF-IDF transformer. We achieved the highest F1-score of 0.98 for binary classification using ALBERT, and 0.94 for multi-class classification—distinguishing between hateful, supportive, and neutral categories—using the Random Forest model.

1 Introduction

The prevalence of hate speech on online platforms poses significant challenges. Yet, the accurate detection of hate speech remains an intricate task, primarily due to the thin line differentiating it to offensive language. Many traditional methods are keyword-dependent, focusing on specific hate speech words, which can be limiting since users frequently employ euphemisms or culturally specific slurs that might not be universally recognized. This complicates the automation of hate speech detection and highlights potential shortcomings of keyword-dependent techniques. This research explores standard machine learning techniques, as well as more advanced ones such as neural networks, for more effective hate speech detection, and compares their performance.

2 Related Work

Much of the work in the field of detecting harmful and hateful speech has been done on a benchmarked set of datasets containing posts from social media. Curry, et all [2] compared classical methods such as using Term Frequency - Inversae Document Frequency (TD-IDF) feature extraction with a Support Vector Machine (SVM) or a Multi Layer Perceptron (MLP) classifier, with deep learning methods that employ contextual word embeddings and a Bidirectional Encoder Representations from Transformers (BERT) classifier.

The most common binary classification done in this field attempts to discriminate between hateful and non-hateful speech. Research has also been done in the multi-category setting looking at predicting different ordinal variables such as racism, genocide, and violence. Kennedy, et all [4], employ deep learning strategies in the multi-category setting.

Often the most challenging distinctions lie in the difference between offensive speech and hate speech. Models are trained to predict if text is "normal", "offensive" or "hateful" offer a greater nuance in their classification. Binny, et all developed a popular dataset called Hate Explained to try and answer this exact question [8]

Other areas of research employ similar model techniques but focus on the unification of multiple hate speech datasets that may have different structure or annotation guidelines. Antypas, et all [1] compare the performance on a subset of 13 hate speech datasets to the unification of said subsets. Their goal in doing so was to create a model that is more robust to differences in data and annotation structure.

A common limitation that our project aims to address is in the comparison of a range of different models on a range of different tasks. Using a new dataset and a variety of different strategies, we analyze the question of hate vs not-hate and the multi-class question of hate vs neutral/ambiguously offensive vs supportive. Analysis of these ambigu-

ous cases as well as supportive texts is something we hope to add to the discussion.

3 Data Sets

3.1 Measuring Hate Speech

The dataset we used was the measuring hate speech (MHS) dataset from the Social Sciences Data Lab at UC Berkeley [4]. We decided to use this dataset because of its large size (136,000 examples) across a diverse set of annotators (11,000 annotators), breadth of labels, multiple platforms, and correction of expected annotator bias.

The main column that we used from MHS was the hate_speech_score. This metric is a continuous measure of hate speech, where higher = more hateful and lower = less hateful. The continuous nature of this measure allowed us to compare results of binary classifications, multi-class classifications, and regressions.

A key feature of this dataset is the representation of hate speech. In the real world, as damaging as hate speech is, it is relatively rate when sampling from social media. It is estimated that less than 1% of social media content is hate speech when considered on a binary outcome. This can make training on a random sample of social media posts incredibly difficult due to class imbalance. To combat this imbalance, the MHS dataset was sampled using a pretrained estimator that predicted the likelihood of hate speech presence in a post. Posts that were "relevant and low on predicted hate speech score", "relevant and high on predicted hate speech score", and "relevant and very high on predicted hate speech score" were oversampled heavily. The result is a much more balanced dataset for us to train our classfier that was sampled without looking for keywords commonly associated with hate speech.

Methods

3.2 Preprocessing

For a binary classifcation of "hate" vs "not hate" we split our dataset into two groups based on the value of hate_speech_score (HSS). A score of above 0.5 is considered hateful by the authors and below is considered neutral or supportive. There are 49048 hateful examples and 86283 non-hateful examples.

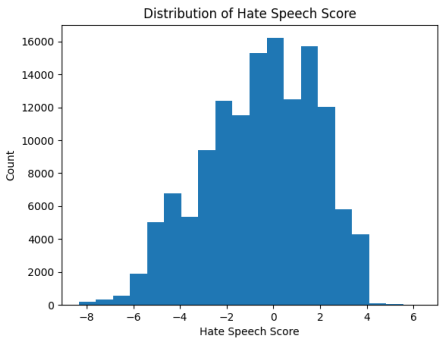

Figure 1: Distribution of continuous hate speech score

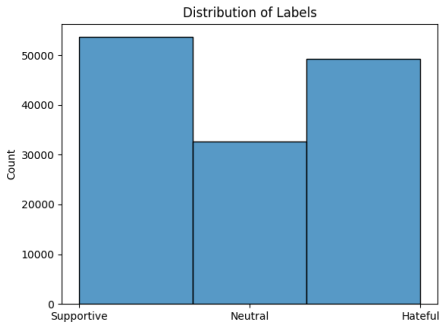

Figure 2: Distribution of continuous hate speech score

In the multiclass setting we split the dataset into three groups, supportive (HSS $<$ -1), neutral or ambigous (HSS $\in$ [-1, 0.5]), and hateful (0.5 $<$ HSS). These splits were decided based on suggestions from the dataset publication [4].

3.3 TD-IDF

For TD-IDF we used the scikit-learn implementation of CountVectorizer. The counts of each unigram, bigram, and trigram are taken then transformed with, TDIDFTransformer. In total, there are 754239 different tokens extracted. In addition CountVectorizer is used to remove the nltk corpus of english stopwords.

3.4 Embedding

We utlized GloVe (Global Vectors for Word Representation) Twitter model[10]. This embedding offered pretrained on Twitter data which was used in our dataset. By utilizing GloVe's contextually-informed vectors, we aimed to enhance our models ability to understand the nuances of the social media based language found in the Berkeley dataset.

3.5 Classical Methods

We used the scikit-learn implementation of a linear support vector machine (SVM) with a C value of 1 and l2 regularization. This model mirrors the architecure used in Curry et all [2]. For use of this linear SVM in the multi-class setting we used the parameter multi_class="ovr." This creates n one-vs-many classfiers and was the recommended multi-class setting in scikit-learn's documentation [9].

In addition to a linear SVM we wanted to test a classifer that would be inherently multiclass when distinguishing between supportive, neutral, and hate speech. We chose to use a small random forest classifier with 5 estimators.

3.6 Long Short Term Memory (LSTM) Neural Network

The LSTM model used is a bidirectional model that mirrors the architecture used in U.B. Mahadevaswamy and P. Swathi [7] for the binary classification and deeper variation of the previous for the multiclass classification task. This model features two bidirectional LSTM's followed by three dense layers and intermittent dropout layers for regularization. For the deeper model we increased the count of LSTM layers by one, and implemented l2 regularization between dense layers to avoid over fitting. All these models were trained for 10 epochs with the Adam optimizer from the keras package. Early Stopping and model check-pointing were implemented also using keras.

3.7 Convolution Neural Network (CNN)

Typically used for image-based tasks, CNNs have also been shown to be effective for text sequence classification [5]. Similar to our LSTM model, this architecture includes an initial GloVe Twitter embedding layer. The model is heavily inspired by the architecture presented in the referenced paper; however, it differs in that it contains two convolutional layers instead of one. Consequently, the model now features two convolutional layers, each followed by max pooling and dropout. The output is then flattened and passed through two dense layers, also incorporating dropout and L2 regularization for improved generalization.

3.8 A Lite Bidirectional Encoder Representations from Transformers (ALBERT)

Large language models have gained significant popularity in natural language processing (NLP) tasks. Among these, BERT (Bidirectional Encoder Representations from Transformers) [3] stands out for its exceptional performance across a variety of tasks. Notably, BERT demonstrates remarkable adaptability to numerous applications when fine-tuned with specific data. However, its computational expense is a notable drawback in its training process. To mitigate this, we opted for ALBERT [6], a streamlined version of BERT that maintains high performance levels while being more computationally efficient. Additionally, the data was tokenized using TensorFlow's pre-trained ALBERT tokenizer.

Results

3.9 Deep Learning Methods

The lSTM model achieved high performance in the binary classification task as demonstrated on Table X. All three metrics achieved scores of 93 - 94 percent. Alternatively, the multi-class task had weaker scores overall with values in the 87-88 percent range. This model was very middle of the pack in all metrics for both tasks and may be a good baseline to compare other models to.

The CNN model has also achieved high performance in binary classification as shown on Table 1. The lower macro-averaged recall is attributable to the 'hate speech' class (class 1) having a lower recall of 0.81. In contrast, the multi-class classification performance is lower as seen on the same table. This is primarily due to the very low precision and recall for the 'neutral' class. While the 'supportive' and 'hate speech' classes achieve mid-80% performance on these metrics, the 'neutral' class scores only 0.61 on both.

ALBERT, in contrast, exhibited exceptional performance across all metrics in binary classification. In fact, it performed the best out of all the models. However, for multi-class classification, it struggled with the 'offensive' class, achieving 75% across all metrics, similarly to the CNN and LSTM models. This impacts the overall macro performance, although it remains superior to that of the CNN and

LSTM.

3.10 Classical Methods

The Support Vector Machine (SVM) also achieves very high performance, ranking just below ALBERT in the binary classification task. While it excels in multi-class performance, this achievement is not directly comparable to inherent multi-class classifiers. This is because the SVM uses three 'one-vs-many' classifiers instead of optimizing a joint objective over the three present classes.

The Random Forest model demonstrates exceptional performance in both binary and multi-class classification tasks. It achieved the highest scores in terms of F1, recall, and precision among all evaluated models for multi-class classification.

Table 1: Performance metrics of every model

	ALBERT	CNN	LSTM	SVM	RF
Precision					
Binary	**0.98**	0.91	0.94	0.97	0.96
Multi	0.85	0.773	0.87	0.94*	**0.94**
Recall					
Binary	**0.975**	0.885	0.95	0.97	0.95
Multi	0.85	0.77	0.88	0.94*	**0.90**
F1					
Binary	**0.98**	0.895	0.94	0.97	0.96
Multi	0.857	0.77	0.87	0.94*	**0.92**

*average across three, one vs many classifiers, not directly comparable to inherent multiclass

4 Discussion

Our LSTM model, with its ability to capture long term dependencies in textual data, performed around middle of the pack in all statistical categories. Our use of pre-trained embeddings allowed it to further capture important, context specific, clues in order to make a more educated prediction of the final class. It performed exceptionally in the binary classification task possibly due to its dependence on keywords present in data in order to differentiate between hate speech and support classes. When it came to the multi-class task it maintained form with slightly lower scores. This speaks to the complexity of the textual data and the nuances present between each target group. Once again the data imbalance in our training set may have hindered the capabilities of our model to generalize well to all three class but such is the nature of data in practice.

The ALBERT model excels in binary classification, likely because is the most complex model among those evaluated, with millions of parameters and extensive pre-training. Fine-tuning these embeddings to specific tasks naturally leads to high performance, as observed in our results.

In binary classification tasks, the Convolutional Neural Network (CNN) exhibits lower recall for the hate speech class, indicating a challenge in correctly identifying hate speech compared to non-hate speech. This issue may stem from the data distribution, which is approximately 65-35, with hate speech being less represented. Despite the substantial volume of data, this imbalance suggests that CNNs may not generalize as effectively as other classical or machine learning methods in this context. This is particularly important to consider when applying these models to real world scenarios where hate speech will make up less than 1% of internet comments.

In multi-class classification tasks the deep learning models we tested show proficiency in distinguishing between hate speech and supportive speech. They do however struggle with the classification of neutral or ambiguous speech. This issue may partly be due from the fact that the neutral speech category is underrepresented, with approximately 15,000 fewer data sentences than other categories. Additionally, the distinction between these ambiguous cases and hate speech is often very challenging to determine. Models cannot rely solely on the presence of keywords as there is often considerable keyword overlap between these classes.

Despite these challenges, the Random Forest model is the most effective model (ignoring SVMs), achieving metrics in the 90s. This superior performance can be attributed to the nature of Random Forests as ensembles of decision trees, which excel in multi-class classification. Furthermore, their method of bootstrap aggregating effectively reduces variance, making them a suitable choice even with the limited representation of offensive speech data.

Our Random Forrest and SVM classifiers make use of TD-IDF unigrams, bigrams, and trigrams. This allows the models to see past just important keywords but also important combinations of words found in each class. A feature importance

study would be an interesting way to learn more about which n-grams are the key to distinguishing these classes.

4.1 Areas for Future Study

In addition to studies of feature importance there are many ways in which we envision this study being extended.

This study was performed on a single hate speech dataset. Much of current research is focused on systems that perform well on multiple unified datasets. By adding in additional datasets that don't necessarily follow the same structure or annotating format as MHS we could develop more robust models.

Furthermore, due to the continuous nature of the hate speech metric analyzed in this study, the MHS dataset presents a rare opportunity to develop a regression system that not only detects hate speech but quantifies how "hateful" a given text is.

5 Contributions

Younes worked on the CNN and ALBERT classification. Mackenzie worked on the LSTM model, and dataset research. Rakesh worked on the SVM/Random Forrest classifiers. All members worked on report equally and helped with preliminary literature review.

6 Conclusion

We trained various machine learning models to detect hate speech using the UC Berkely Social Science Data Lab's hate speech dataset [4]. We see that ALBERT excels in distinguishing hate speech from non-hate speech with an F-score of 0.98. However, the Random Forest model proves to be superior in the multi-class differentiation of supportive, ambiguous, and hateful speech with an macro F-score of 0.92. The classification of text that is ambiguous and offensive but not "hateful" proved to be the most challenging task. Overall, classic machine learning models perform better in multi-class classification tasks compared to deep learning models, which struggle to categorize text which is neutral or ambigious.

7 Acknowledgements

The code to fine-tune ALBERT was inspired by The distillBERT text classification Hugging Face tutorial (https://colab.research.google.com/github/peterbayerle/huggingface_notebook/blob/main/distilbert_tf.ipynb).

8 Literature Cited

References

[1] Dimosthenis Antypas and Jose Camacho-Collados. Robust hate speech detection in social media: A cross-dataset empirical evaluation, 2023.

[2] Amanda Cercas Curry, Gavin Abercrombie, and Verena Rieser. Convabuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational ai, 2021.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Chris J. Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application, 2020.

[5] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[7] U.B. Mahadevaswamy and P. Swathi. Sentiment analysis using bidirectional lstm network. *Procedia Computer Science*, 218:45–56, 2023. International Conference on Machine Learning and Data Engineering.

[8] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. *CoRR*, abs/2012.10289, 2020.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.