Comparative study of BERT models in Hate Speech Detection
HateSpeech Hunter

Sankalp Talankar, Sakshee Bahadekar, Pujan Kothari, Ratna Prabha Bhairagond
{sankalp.talankar,sbahadekar,p.kothari,r.bhairagond}@ufl.edu

Abstract

This study is based on the domain of hate speech detection, and its primary aim is to perform a comparative study of different BERT models. It utilizes a dataset of social media posts annotated for hate speech, and trains and evaluates various BERT models, like RoBERTa, ALBERT, DistilBERT, and ELECTRA. The study compares the models based on metrics such as accuracy, F1 score, precision, and recall. The results show that certain BERT models outperform others in hate speech detection, and the study discusses the architectural differences that produce varying results.

1 Introduction

The rise of hate speech on various social media platforms has become a significant cause for concern in recent years. This has triggered the need for effective hate speech detection tools. It is nearly impossible to manually monitor and identify hate speech on social media sites given the enormous volume of content that is produced there every day. Effective hate speech detection systems that can swiftly recognize and label hateful content are now required as a result of this. For instance, during the 2016 U.S. presidential election, there were numerous instances of hate speech being used by supporters of both candidates. This included derogatory comments about immigrants, racial and religious minorities, and members of the LGBTQ+ community. Natural language processing (NLP) techniques have proved to be a useful tool in the identification of such hate speech.

Hate speech detection tools can scan vast amounts of text and find patterns and signs of hate speech by applying NLP approaches, assisting in reducing its negative impacts. Deep learning models such as transformers have shown very good performance for natural language processing tasks including hate speech detection. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that utilizes a transformer-based architecture and has the ability to capture complex linguistic relationships and context. There are many different types of BERT models that have been developed, each with its own modifications and advantages.

In this study, we will explore how four popular BERT models perform the task of hate speech detection namely RoBERTa, ALBERT, DistilBERT, and ELECTRA. RoBERTa is a robustly optimized BERT approach that uses larger datasets and more extensive pre-training than the original BERT model. ALBERT is a lite version of BERT that utilizes parameter-sharing techniques to reduce the model size and improve efficiency without sacrificing performance. DistilBERT is another lite version of BERT that uses knowledge distillation to transfer knowledge from a large BERT model to a smaller one. Finally, ELECTRA is a novel BERT-based model that uses a generative adversarial network (GAN) approach to pre-training, resulting in improved performance on downstream tasks.

The choice of the BERT model depends on a lot of factors like the size of the datasets, the type of hate speech, and the level of accuracy and efficiency. The aim of this study is to conduct an evaluation and comparison of the performance of different BERT models for hate speech detection to identify the most ideal model for the task. The results of this study will provide insights into the strengths and weaknesses of each model and inform the development of more effective hate speech detection systems. A system like this could be utilized by social media platforms, government organizations, and other entities to mitigate the harmful effects of hate speech. Furthermore, this study could contribute to the field of natural language processing and provide new insights into the capabilities of BERT models for hate speech detection.

2 Literature Review

This section aims to provide an overview of previous research done on hate speech detection. It will also emphasize the gaps in current research and identify how our research can fill them.

Hate speech detection is a very important application of NLP and groundbreaking progress has been made using Attention-based LSTMs [De la Pena Sarracén et al., 2018] and Transfer Learning Approaches for BERT models [Mozafari et al., 2019]. Another implementation of BERT for hate speech detection is HateBERT where BERT was retrained for detecting abusive language in English. The resulting model outperformed the general BERT model and while these results are promising, this study employed an uncased BERT model [Caselli et al., 2020].

Within the domain of hate speech detection, the presence of capital letters strongly suggests the presence of hate speech and hence the gap between cased and uncased BERT models is a blank space that requires further investigation.

Fine Tuning BERT using a Genetic Algorithm-based approach [Madukwe et al., 2020] has outperformed the general BERT and also reduced the time and cost needed for designing such fine-tuning methods. However, another gap observed in these BERT-based studies is that they were carried out for the general BERT model. BERT variations like RoBERTa, ALBERT, DistilBERT, and ELECTRA are highly optimized and have proved to be very strong tools for NLP tasks. A study of DistilBERT fine-tuned for the specific task of hate speech detection provided insights into how fine-tuning can lead to improved performance as compared to general BERT [Kumar et al., 2020].

The question that now arises is, can similar fine-tuning be done for other models; and, how do these models compare when pitted against each other?

3 Research Methodology

The aim of this study was to conduct an analysis of the performance of various BERT models for hate speech detection. The “Measuring Hate Speech" Dataset by UC Berkeley’s D-Lab is used to gain insights into public sentiment and attitudes towards the pandemic. A dataset of 135,000 tweets was collected using relevant hashtags and keywords and preprocessed using standard techniques such as stop word removal, punctuation removal, tokenization, and stemming. For detection, four pre-trained BERT models were compared, specifically RoBERTa, ALBERT, DistilBERT, and ELECTRA. These models were fine-tuned on a large corpus of social media text, and trained and validated on a 70/30 train-test split. Accuracy, Precision, Recall, and F1-score were used as evaluation metrics for the model. The methodology used in this study provides a rigorous and comprehensive approach to analyzing sentiment in social media data using various BERT models and can serve as a base for identifying how BERT models work and how they can be used in different applications.

3.1 Algorithms/Models

This section will describe the architecture and working principles of the BERT-based models (RoBERTa, ALBERT, ELECTRA, and DistilBERT) that are used in the study.

3.1.1 RoBERTa

RoBERTa is a language model based on the BERT architecture which has gained significant popularity in the field of natural language processing, particularly in hate speech detection. RoBERTa is optimized for certain tasks like hate speech identification since it has been pre-trained on vast amounts of text data. The maximum sequence length and learning rate are two factors that must be carefully taken into account to ensure optimal performance [Liu et al., 2019].

In this study, we created a RoBERTa model for hate speech detection by leveraging the TFRobertaModel from the transformers library and fine-tuning it on the Measuring Hate Speech dataset. Our RoBERTa model utilized two input layers, one for the input IDs and the other for the attention masks. It employed a transformer-based encoder with a multi-head self-attention mechanism to analyze and encode input text sequences into contextualized representations. We set the output dimension to the first token of the last layer’s output and passed it through a dense layer with 128 units and a dropout layer to prevent overfitting. Finally, we used a dense layer with two units and a softmax activation function as the output layer to predict the probability of each class.

To prepare the data for our model, we applied various preprocessing techniques such as lowercasing, removing non-alphanumeric characters, removing stop words, stemming and lemmatization, and handling imbalanced classes using the RandomOverSampler from the imblearn library. Additionally, we used Word2Vec from the gensim library to generate word embeddings. We split the Measuring Hate Speech dataset into a training set of 70% and a testing set of 30%. The preprocessed data were then trained on three epochs and a learning rate of 1e-5. We then used precision, recall, F1 score, and accuracy metrics to evaluate the performance of our model's output.

3.1.2 ALBERT

ALBERT - A Lite BERT is a variation of BERT however what sets it apart from the other variants is the number of learnable parameters. These parameters are the weights or biases that are optimized throughout the training process using three techniques [Lan et al., 2019]. The first is factorized embedding parameterization. The efficiency of the model is improved by using different sizes for the WordPiece embedding and the hidden layer. Having the same size can lead to inefficient use of resources, especially for larger vocabulary sizes. Using different sizes and embedding them to a lower dimensional space ensures that the number of parameters is reduced.

The second is cross-layer parameter sharing; this technique shares the same number of parameters across all the layers of the model, hence reducing the number of parameters that are shared across multiple layers and increasing efficiency. The third technique is inter-sentence coherence loss, in order to understand the relationships between two sentences, Albert tries to predict whether the two sentences are in order or not. It does so by employing a sentence-order prediction (SOP) loss which outperforms NSP method used in the general BERT model.

The WordPiece embedding and the transformer encoder encodes the text into token embeddings. Relationships between the sentences are established using feedforward neural networks and self-attention mechanisms. This processing is done in the encoder layers following which the output is sent to a pooling layer and then a final layer that predicts the target label for the given input. This facilitates a reduction in training time while maintaining performance.

An ADAM Optimizer is employed with a learning rate of 5e - 5. It encourages the gradients to converge to a point of minima for the loss function more quickly. The loss is calculated as the cross entropy loss between the actual values and the ones predicted by the model. A softmax activation function is used that considers the possible outcomes and generates the corresponding probability distribution. This model was trained on the dataset for 10 epochs with a 0.7 to 0.3 training-testing split on the dataset and the results were evaluated.

3.1.3 ELECTRA

The ELECTRA model stands for "Efficiently Learning an Encoder that Classifies Token Replacements Accurately," and it is a pre-training approach for language modeling based on transformer architecture. Unlike other pre-training models, such as BERT, which uses a masked language modeling (MLM) approach to predict masked tokens, ELECTRA uses a novel approach called "discriminator pre-training." [Clark et al., 2020]

ELECTRA is a technique for pre-training models that involves training two transformer models - the generator and the discriminator. The generator is responsible for substituting tokens in a sequence and is trained as a masked language model. The focus is on the discriminator, which attempts to recognize the tokens that were replaced by the generator in the sequence [Clark et al., 2020]. The ELECTRA model is good at understanding the relationships between words and phrases, which makes it a better model for understanding natural language.

In discriminator pre-training, the model learns to differentiate between "real" input tokens and "fake" tokens that are generated by replacing some input tokens with random tokens. The model is trained to maximize the probability of correctly classifying whether a token is real or fake [Clark et al., 2020]. This approach allows the model to learn more efficiently because it only needs to predict whether a token is real or fake, which is a binary classification task, rather than predicting the exact token.

In this research study, we fine-tuned the ELECTRA model for hate speech detection. To overcome memory and CPU limitations, we utilized the ELECTRA-small model in conjunction with the ElectraForSequenceClassification model and ElectraTokenizerFast tokenizer from the Hugging Face library. We trained our model on the 'Measuring Hate Speech' dataset, which includes a 'hate_speech_score' column containing both positive and negative values to represent positive or negative sentiment. To convert these values to binary scores, we preprocessed the dataset by stemming, lemmatizing, removing stop words and unnecessary columns, and adding a new column called ’label’ which represented the binary score for ’hate_speech_score’.

The preprocessed dataset was then split into a training set and a test set with a ratio of 70:30. The labels in both sets were examined, and it was found that the dataset was balanced. The training dataset was then converted into a TrainerDataset object, which was fed to the Trainer class along with the evaluation dataset.

The model was trained for 3 epochs with a batch size of 32, using a learning rate of 5e-5 and a dropout rate of 0.1. Finally, evaluation metrics such as accuracy, precision, recall, and F1 score were computed.

3.1.4 DistilBERT

DistilBERT is a smaller and faster version of the BERT transformer model [Sanh et al., 2019]. It uses a technique called distillation to compress the original BERT model by removing unnecessary layers and parameters while still preserving its performance. This results in a model that is 60 percent smaller and 40 percent faster than BERT while retaining 97 of its performance on certain tasks. This makes it more efficient on training small datasets or simple datasets. It is also better where the time-to-respond time requires is less.

The data preprocessing has been done using techniques such as lower casing of texts, removing all punctuations, removing stopwords, and finally performing lemmatization. The DistilBERT model is loaded from the pre-trained checkpoint using the pre-trained method. The input to the model is given in the form of a sequence of tokens using DistilbertTokenizer represented by their token IDs (input ids) and an attention mask that indicates which tokens should be taken care of paid attention to during model training and which ones should be avoided(attention mask).

The model will give an output of a sequence of hidden states, one for each token that was a part of the input sequence. These hidden states are important for sentiment classification as they will be used to perform the classification. This hidden state is then passed through a dropout layer and a linear layer, which outputs a score for each possible class, indicating the predicted sentiment of the input sentence. The base DistilBERT model is used with the Adam optimizer. The learning rate is 2e-5 with the batch size being 16. We used a dropout rate of 0.3. The training was done for 3 epochs and then the performance was analyzed.

3.2 Dataset

The "Measuring Hate Speech" dataset, developed by UC Berkeley D-Lab, is a comprehensive dataset containing 135,556 rows of annotated social media comments. It serves as a valuable resource for training hate speech detection models. The dataset includes a primary outcome variable, the "hate speech score," which ranges from -8 to 6, with higher scores indicating more hateful content. Additionally, it has 10 categories representing different aspects of hate speech, such as sentiment, disrespect, insult, violence, and dehumanization, among others. The dataset also provides information on 8 target identity groups and 42 target identity subgroups, offering a diverse range of data to train the model on [Kennedy et al., 2020].

One of the reasons for choosing this dataset is its large size and extensive annotation, with comments annotated by 7,912 annotators. This ensures a robust and diverse set of labels for training the model. Moreover, the inclusion of various target identity groups and subgroups in the dataset allows for a comprehensive analysis of hate speech related to different identity categories, which can provide insights into patterns and trends of hate speech targeting specific communities.

In the initial analysis, other characteristics of the data are also considered, such as text length, average word length, common stopwords, capital words, the frequency distribution of common words, bigrams, trigrams, and intentional spelling mistakes. These features help capture different linguistic patterns and techniques used by hate communities, making the dataset more reliable for training a BERT model for hate speech detection.

3.2.1 Data Preprocessing

Data preprocessing techniques play a critical role in natural language processing, particularly in hate speech detection using BERT models. In this research study, we applied various preprocessing techniques to the dataset, including cleaning, stop word removal, stemming, lemmatization, label encoding, handling imbalanced classes, and word embedding. Firstly, we cleaned the text and removed unnecessary characters and symbols which helped us to standardize the data and make it consistent, which is important for training BERT models. Secondly, we did stop words removal, this helped us to reduce the dimensionality of the data and improve the quality of the features by removing words that did not provide significant information. We also applied stemming and lemmatization, on the other hand, which was used to further simplify the vocabulary and normalize the text. We converted the hate speech score ranging from -8 (not hate speech) to 6 (very hate speech) into binary labels (0 for not hate speech and 1 for hate speech) using a threshold of 0. This step was important because machine learning models, in particular BERT models are more effective when the vocabulary is reduced, and the text is represented in a consistent format.

Next in our models, we applied label encoding which converted the target variable into a numerical format that could be fed into machine learning models. This was important because models require numerical data to learn and make predictions. Handling imbalanced classes is crucial for improving the performance of the model, as models trained on imbalanced datasets tend to perform poorly on the minority class. Finally, word embedding was used to represent the text in a numerical format that can be fed into BERT models. Word embeddings capture the semantic meaning of words and their relationships with other words in the text, making them effective for training machine learning models.

These preprocessing techniques have significantly impacted the quality and performance of hate speech detection models based on BERT architectures. The proper implementation of these techniques helps simplify the vocabulary, balance the distribution of classes, and represent the text in a numerical format that the models can understand, resulting in improved performance and quality of the models.

3.2.2 Data Visualization

These are some of the insights that we have derived from the data, and they are visually represented below. Figure 1 represents the hate speed score distribution in the dataset.

Figure 2 shows the most commonly occurring words in the dataset.

3.3 Fine Tuning

The fine-tuning process was geared toward finding the sweet spot/trade-off between two extremities. This section discusses the hyperparameters considered and their impact on the training process. Optimal values of the hyperparameters have been presented in Table 2 of the subsequent Results and Discussions section.

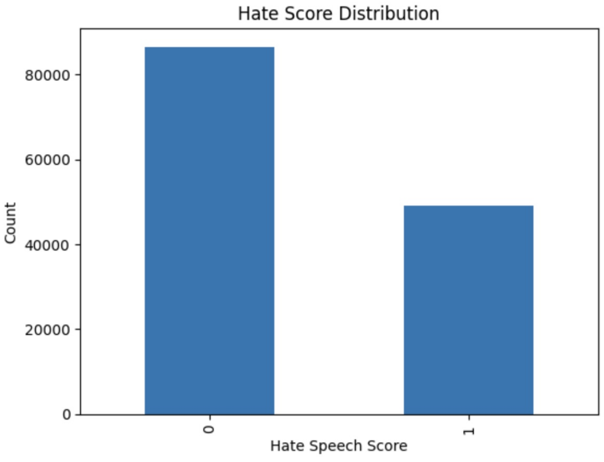

Figure 1: Hate speech score distribution

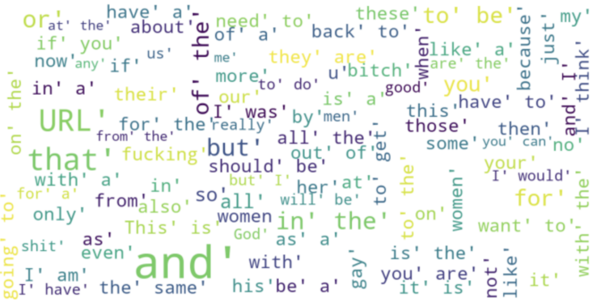

Figure 2: Most common words in the dataset.

- Learning Rate - A very high learning rate leads to an unstable model as the large gradients might cause the model to overshoot the optimal value. However, a small rate would take too long to reach the optimal value.

- Number of samples - Having a very small number of samples gives very little context and is not enough to draw out all the intricacies. A large number of data samples is highly taxing on the computational resources and takes up a lot of time.

- Number of epochs - A very small number of epochs leads to a sub-optimal model, a very large one has the same issues as the case with a large sample size as discussed above.

- Batch Size - A large batch size provides for diversity in the training samples and gives a smoother optimization curve. On the downside, this is very heavy on resources, and memory constraints make it impractical.

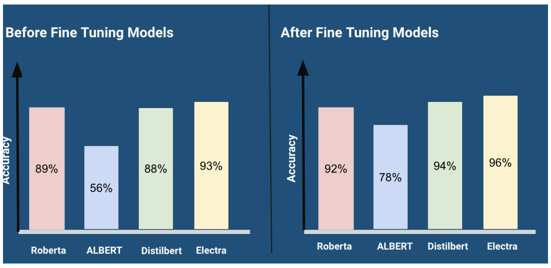

Figure 3: Accuracy of models before and after fine-tuning

3.4 Ground Truth

In our hate speech detection experiment, the term "ground truth" refers to the accurate classification of each text sample in our dataset as either hate speech or not hate speech. This classification applies to both the training and test data.

For our research, we used the "Measuring Hate Speech" dataset provided by UC Berkeley's D-Lab as mentioned in Section 3.2. This dataset comprises annotated tweets labeled for hate speech. The ground truth in our experiment was based on the "hate speech score" assigned by 7,912 annotators to each comment. The annotators used a set of labeling guidelines provided to them (Figure 4) and came from diverse demographics, including gender, race/ethnicity, age, education level, and political ideology. This diversity ensured the dataset represents a wide range of perspectives and experiences. The large size and diversity of annotators make this dataset a valuable resource for training and evaluating hate speech detection models [Kennedy et al., 2020].

The paper Kennedy et al., 2020 reports interrater agreement scores using Fleiss' Kappa coefficient based on three randomly selected annotators for both tasks. For the hate speech task, the authors report a Fleiss' Kappa of 0.56, indicating moderate agreement, while for the targeted attack task, the Fleiss' Kappa was 0.65, indicating substantial agreement.

Each tweet in the dataset has a hate speech score ranging from -8 (not hate speech) to 6 (very hate speech). We converted these scores into binary labels (0 for not hate speech and 1 for hate speech) using a threshold of 0.

We split the dataset randomly into a training set and a test set with a test size of 0.3. We used the ground truth labels of both sets to train and evaluate our model for hate speech detection, respectively.

Figure 4 illustrates the guidelines followed by the annotators.

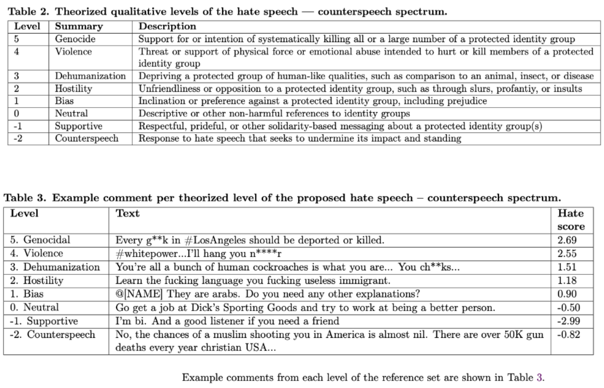

Table 2. Theorized qualitative levels of the hate speech — counterspeech spectrum.

Level	Summary	Description
5	Genocide	Support for or intention of systematically killing all or a large number of a protected identity group
4	Violence	Threat or support of physical force or emotional abuse intended to hurt or kill members of a protected identity group
3	Dehumanization	Depriving a protected group of human-like qualities, such as comparison to an animal, insect, or disease
2	Hostility	Unfriendliness or opposition to a protected identity group, such as through slurs, profantiy, or insults
1	Bias	Inclination or preference against a protected identity group, including prejudice
0	Neutral	Descriptive or other non-harmful references to identity groups
-1	Supportive	Respectful, prideful, or other solidarity-based messaging about a protected identity group(s)
-2	Counterspeech	Response to hate speech that seeks to undermine its impact and standing

Table 3. Example comment per theorized level of the proposed hate speech – counterspeech spectrum.

Level	Text	Hate score
5. Genocidal	Every g**k in #LosAngeles should be deported or killed.	2.69
4. Violence	#whitepower...I'll hang you n****r	2.55
3. Dehumanization	You're all a bunch of human cockroaches is what you are... You ch**ks...	1.51
2. Hostility	Learn the fucking language you fucking useless immigrant.	1.18
1. Bias	@[NAME] They are arabs. Do you need any other explanations?	0.90
0. Neutral	Go get a job at Dick's Sporting Goods and try to work at being a better person.	-0.50
-1. Supportive	I'm bi. And a good listener if you need a friend	-2.99
-2. Counterspeech	No, the chances of a muslim shooting you in America is almost nil. There are over 50K gun deaths every year christian USA...	-0.82

Example comments from each level of the reference set are shown in Table 3.

Figure 4: Example comment from each level of the reference set. (Source: Kennedy et al., 2020)

4 Results and Discussions

After data pre-processing and making it ready for training, we used the four models i.e. RoBERTa, ALBERT, ELECTRA, and DistilBERT to test which will perform the best. Our tests ended up giving the following accuracies.

Model	Accuracy	Precision	Recall	F1-Score
RoBERTa	92.21	92.97	91.47	92.53
ALBERT	78	61	78	68
DistilBERT	94.43	94.03	93.91	93.97
ELECTRA	96.35	96.35	96.35	96.35

Table 1: Accuracy scores of BERT models

The RoBERTa model takes a lot of time to train. The desired accuracy has not been reached, as the RoBERTa model is bulky the appropriate convergence cannot be reached without a training time trade-off. The same applies to ALBERT. DistilBERT does train faster and provide better accuracy but it is still not the best. ELECTRA provides the best trade-off between training time and accuracy.

We also trained these models using different hyperparameters, including learning rates, epochs, batch size, and optimizer. For learning rates, we experimented with values of 1e-6, 1e-4, and the best-performing rate of 1e-5. Similarly, for epochs, we tried values of 1, 2, 3, 4, and 5, with the best performance obtained at 3 and 4 epochs. We also varied the batch size and found that a batch size of 16 gave the best results, followed by a batch size of 32. Finally, we used the Adam optimizer for all experiments.

After training, the ELECTRA model achieved an accuracy of 0.9635 and an F1 score of 0.9635 after

3 epochs as shown in Table 1. The training process took 656.4294 seconds with 41.302 samples per second and 5.163 steps per second.

Our results showed that ELECTRA and DistilBERT outperformed the other models, with ELECTRA achieving the best overall performance. These findings demonstrate the importance of careful hyperparameter tuning when using transformer-based models for NLP tasks.

Hyperparameter	Value Options	Best Value
Learning Rate	1e-6, 1e-4, 1e-5	1e-5
Epochs	1, 2, 3, 4, 5	3, 4
Batch Size	16, 32	16
Optimizer	Adam	Adam

Table 2: Hyperparameters used in the research paper

5 Key Takeaways

Data preprocessing is very important for the performance of NLP models. BERT models are no exception - in fact, text preprocessing is especially important for BERT models due to their large size and complexity. Proper text preprocessing can improve the accuracy of BERT models by ensuring that they are trained on high-quality input data. This includes tasks such as removing irrelevant characters, converting text to lowercase, and handling special characters or symbols.

However, text preprocessing alone may not be enough to ensure optimal performance for BERT models. For example, training large models like ALBERT and RoBERTa requires significant GPU memory to avoid run-out-of-memory errors. These models have millions of parameters and require large amounts of computational power to train effectively. Thus, it is important to carefully manage GPU resources and potentially use distributed training methods to optimize training performance and avoid resource constraints.

While we initially hypothesized that RoBERTa would perform the best due to its larger pre-training dataset, our findings showed that ELECTRA outperformed all other models in terms of accuracy, precision, recall, and F1 score. We attribute ELECTRA's superior performance to its unique pre-training process, which involves training two models - a generator and discriminator - simultaneously. This approach allows for more efficient use of pre-training data and has been shown to lead to better performance on downstream tasks.

Consider a sentence “I don’t think gay people should be allowed to get married”. A traditional BERT model will pre-process the sentence in order and may only learn to associate “gay people” with “married”. The model makes these associations without fully understanding the context. In ELECTRA, however, the generator is trained in a way that enables it to form plausible sentences by replacing “gay people” with “same-sex” or “LGBTQ+ individuals”. The discriminator is then trained to distinguish between the real and generated sentences which inherently requires it to fully understand the context between the tokens. Additionally, ELECTRA’s smaller model size further contributes to its success.

On the contrary, ALBERT has consistently produced lower accuracy results, however, it is imperative to note that it is a larger model, and hence the training process was overwhelming for the server. During the process of training the dataset had to be broken down into a smaller subset and training was done over fewer epochs. These parameters have a direct impact on the result of the model, and with more computational resources the results are bound to be better.

6 Conclusion

We were able to ascertain the best and worst performing models and also identify the most important metric while evaluating them. Our research demonstrates that the ELECTRA model is a promising choice for hate speech detection, and suggests that researchers and practitioners should consider a range of model architectures and evaluation metrics when developing natural language processing solutions. Our analysis also revealed that recall emerged as the most important metric in hate speech detection. This is because correctly identifying all true positive hate speech cases is crucial, even if it means that some false positives are included. Our findings provide valuable insights for researchers and practitioners in the field of natural language processing, highlighting the importance of exploring different model architectures and evaluating performance on specific metrics for specific tasks.

7 Future Work

Future research in hate speech detection should focus on developing models that can effectively identify and classify hate speech in multiple languages, as well as classify it into various subcategories such as gender, race, religion, and sexual orientation. Additionally, integrating multiple modalities like audio and video into hate speech detection can enhance its accuracy by providing additional contextual cues. Developing such models will be crucial in combatting the rise of hate speech and promoting more tolerant and inclusive social media platforms

8 Github Link

https://github.com/Hate-Speech-Hunters/Hate-Speech-Detection

References

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, and Michael Granitzer. 2020. Hatebert: Retraining BERT for abusive language detection in english. *CoRR*, abs/2010.12472.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Gretel Liz De la Pena Sarracén, Reynaldo Gil Pons, Carlos Enrique Muniz Cuza, and Paolo Rosso. 2018. Hate speech detection using attention-based lstm. *EVALITA evaluation of NLP and speech tools for Italian*, 12:235.

Chris J. Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *CoRR*, abs/2009.10277.

Abhinav Kumar, Sunil Saumya, and Jyoti Prakash Singh. 2020. Nitp-ai-nlp@ hasoc-fire2020: Fine tuned bert for the hate speech and offensive content identification from social media. In *FIRE (Working Notes)*, pages 266–273.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kosisochukwu Judith Madukwe, Xiaoying Gao, and Bing Xue. 2020. A ga-based approach to fine-tuning bert for hate speech detection. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2821–2828.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. *CoRR*, abs/1910.12574.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.