Enhancing Efficiency in Hate Speech Detection

Joseph Thomas Saarthak Singh Subham Satapathy Vidit Agrawal

1. Abstract

This project aims to advance the efficiency of hate speech detection using large language models (LLMs). Hate speech online can foster toxicity and harm, necessitating accurate detection. However, existing efforts emphasize performance over efficiency. Leveraging recent progress in streamlined LLMs, this research explores enhancing detection speed and reducing size while preserving accuracy. The methodology employs strategies like pre-training, and fine-tuning LLMs to optimize language understanding, combined with three seminal hate speech datasets - HateXplain, Hate Speech 18, and Toxigen - encompassing diverse data sources and annotations. Following thorough exploratory analysis, feature engineering, data was prepared over which current benchmarks and baselines were established using popular models.

Subsequently, post pretraining custom Bert model, fine tuning and compression techniques were explored to reduced parameters and accelerated inference without substantial performance decline. Comparative evaluation analyzes tradeoffs, with target metrics resulting in the selected model on the all three aggregated datasets with pretrained BERT teacher and TinyBERT student achieving the best balance of performance, efficiency and generalization. With online hate rising, this research tackles critical detection challenges. It aspires to balance interpretation, generalization, and efficiency, enabling models accessible across various platforms. The project contributes significantly towards mitigating online toxicity through prompt and robust hate speech flagging at scale. With ethical considerations around biases accounted for, the anticipated outcomes will make state-of-the-art detection achievable on low-resource systems.

2. Introduction

Online hate speech is a growing threat, fostering toxicity and hindering constructive dialogue. Accurate detection is crucial, but current methods prioritize performance over computational efficiency. We assert that optimizing inference time and model size is equally vital for feasibility and accessibility in various applications.

In this paper, we utilize model compression techniques to significantly reduce parameters, speeding up predictions without compromising accuracy. Custom classifier architectures are created by fine-tuning large pre-trained language models. We conduct comparative evaluations on three diverse hate speech datasets to assess optimization approaches.

This work is the first to simultaneously enhance efficiency and effectiveness in hate speech detection. We explore trade-offs between model performance and computational requirements, presenting optimized models that enable swift and robust flagging of online toxicity in resource-limited systems. Our research marks significant progress in combating online harm at scale through efficiency improvements in modern deep learning.

The core contributions of this paper are three-fold:

- Following preprocessing, we provide a variety of pre-trained models with dataset and architecture variations, available for public use on Hugging Face. We also employ well-known neural models as benchmarks.
- We construct optimized and finetuned architectures leveraging state-of-the-art model compression techniques.
- Conducted a comparative evaluation of optimization approaches, assessing both accuracy and efficiency. Comparde techniques on individual datasets, cross-reference across multiple datasets with public benchmarks, and validated generalization using a model proposed on aggregated datasets.

The outcomes of this work will facilitate accessible and ethical integration of advanced hate speech detection into diverse practical deployments through pioneering focus on efficiency enhancements.

3. Related Work

In recent hate speech detection research, several noteworthy approaches have emerged, each emphasizing distinct strategies and datasets to enhance model performance. “HateBERT: Retraining BERT for Abusive Language Detection in English”[14] involves retraining BERT, a leading

language model, for English abusive language detection, leveraging a dataset derived from Reddit comments (Rale). This highlights the benefits of retraining BERT and its potential performance boost.

Another paper "Hate speech detection using static BERT embeddings"[24] employs static BERT embeddings with the Ethos dataset, processed through a deep neural network (DNN) to detect hate speech. This research underscores the value of pre-trained models like BERT. Additionally, "A New Hate Speech Detection System based on Textual and Psychological Features"[11] introduces a hate speech detection system using both textual and psychological features, drawing from Twitter data. It addresses data imbalance and evaluates various models, with Long Short-Term Memory (LSTM) and BERT showing promise.

Furthermore, the paper "Masked Rationale Prediction for Explainable Hate Speech Detection"[18] achieves state-of-the-art results in explainable hate speech detection on the HateXplain dataset. This involves pre-finetuning data and applying random masking before.

Further, "Deep Learning Models for Multilingual Hate Speech Detection"[12] introduces a novel approach for multilingual hate speech detection, combining various embeddings and models. It excels in zero-shot, low-resource, and high-resource languages, providing an adaptable solution to combat online hate speech. "Learning from the Worst: Dynamically generated datasets to improve online hate detection"[28] innovatively enhances online hate detection through dynamically generated datasets and RoBERTa. Surprisingly, models trained on increasingly challenging data perform better, highlighting the potential for real-world hate detection improvement.

In the paper, "Detecting Online Hate Speech Using Context-Aware Models"[15] combines logistic regression, bi-directional LSTM models, and ensembles for online hate speech detection. The Max Score ensemble excels in recall and F1 score, showcasing its effectiveness. "Improving Hate Speech Detection with Deep Learning Ensembles"[30] introduces CNN ensemble models with diverse weight initializations, achieving significant F1 score improvements. It highlights the potential of deep learning ensembles for enhancing online hate speech detection.

In "DeL-haTE: A Deep Learning Tunable Ensemble for Hate Speech Detection"[22], DeL-haTE employs word embeddings and a combination of CNN, RNN, and fully connected models. It outperforms individual models with notable gains in hate recall and F1 score, particularly when weakly supervised using GAB data. This research emphasizes the promise of tunable ensembles and weak supervision in advancing hate speech detection. "ToxiGen: A Machine-Generated Hate Speech Dataset"[16] introduces ToxiGen, a unique dataset for adversarial and implicit hate speech detection. It employs demonstration-based prompting, with ALICE generating challenging content. Impressively, 90.5% of machine-generated examples appear human-written. This research advances hate speech detection, focusing on subtle and implicit online hate speech.

The papers "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations"[20], "MobileBERT: a Compact Task-Agnostic BERT Task-Agnostic BERT for Resource-Limited Devices"[26], and "TinyBERT: Distilling BERT for Natural Language Understanding"[17] represent cutting-edge efforts in streamlining deep learning models for language processing. They prioritize efficiency without sacrificing performance. ALBERT employs parameter-reduction techniques, reducing memory usage and speeding up training. MobileBERT adapts BERT for mobile devices, balancing size, speed, and accuracy. TinyBERT distills BERT's insights into a smaller model. These innovations enhance NLP model scalability and accessibility across various platforms.

Complementing these advances is "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter"[25], which streamlines BERT into a more compact and efficient variant suitable for real-time applications, mitigating prior computational drawbacks. A different approach is showcased in the exploration of CNNs for sentence classification in "Convolutional Neural Networks for Sentence Classification"[19], which departs from transformer-based models. This research applies convolutional layers to text data, utilizing word embeddings and filters to extract key features through max-pooling, culminating in a softmax layer for classification. The study's findings contribute to a broader understanding of NLP, suggesting that CNNs can also effectively discern linguistic patterns for sentence classification. Together, these papers illustrate the diverse methodologies emerging in the field of NLP, each pushing the boundaries of what is possible in processing and understanding human language through machine learning.

The study, "Hate Speech Detection on Twitter Using BERT Algorithm"[23], investigated the use of the BERT language model to detect hate speech in English and Indonesian tweets. The study found that BERT achieved an accuracy of 78.69% for the English dataset and 68% for the Indonesian dataset.

Furthermore, "BERT-based ensemble learning for multi-aspect hate speech detection"[21], combined the BERT language model with other Bi-LSTM- and Bi-GRU-based models to improve its performance in textual hate speech detection. The study found that the ensemble learning approach reduced the number of misclassified instances and thus improved the precision of hate speech detectors. The study "Interpretable Multi Labeled Bengali Toxic Comments Classification using Deep Learning"[13] used transfer learning by using BERT and CNN-BiLSTM to classify

Bengali toxic comments. The study found that the proposed model outperformed a fine-tuned pre-trained transformer model in both binary and multi-label classification tasks.

These studies collectively underscore the significance of advanced models and techniques in addressing hate speech detection across diverse languages and contexts, emphasizing the continual evolution of the field. However, there is still room for improvement, as the performance of these models varies depending on the language and dataset used.

4. Data

We processed three key hate speech datasets—Hate Speech 18, HateXplain, and Toxigen. Recognizing that hate speech hinges on context and specific words may introduce bias, we prioritize model interpretability. Combining these datasets, our aim is to cover diverse examples of hate speech across various contexts, targets, and linguistic styles.

4.1. Hate Speech 18

In this study, we leverage the "Hate Speech 18"[2] dataset, a seminal corpus comprising approximately 11k English text, with Stormfront as the primary resource. This dataset, developed for the SemEval-2019 Task 5, categorizes tweets into three distinct classes: hate, noHate, idk/skip and undefined.

The dataset's heterogeneity in tweet length significantly influences the feature dimensionality, posing unique challenges and opportunities for textual analysis and model development. Moreover, the dataset's extensive popularity, as indicated by *paperwithcode.com i*s a testament to its widespread utilization, providing further validation of its usage.

Preprocessing: Before model implementation, a comprehensive preprocessing protocol was employed to enhance data quality and consistency. This involved the removal of extraneous metadata, the transformation of textual content to align with NLP requisites (including tokenization, stopwords and punctuation elimination), and rectification of missing data. Further, we truncated the length to 256 char (less than 2% of data average) while pretraining in regards to memory constraint. We ignored undefined labels, kept the 'hate' label as '1' and the rest as '0'.

The research culminates in the development of a binary classification model, tasked with distinguishing between hateful and non-hateful tweets.

4.2. HateXplain

In this study, we employ the HateXplain dataset [3], which consists of approximately 9,491 English text from Twitter and Gab, labelled by multiple annotators. Tweets have the label as hate, normal or offensive. It helps with additional generalization, and ensures that offensive viewpoints are not misclassified as hate-speech.

Preprocessing: Our preprocessing approach [8] for the HateXplain dataset includes tokenization, lowercasing, and the removal of stop words and punctuation. Various labels were combined based on max count. Additionally, we conducted a careful selection, prioritizing those with clear hate speech annotations and comprehensive explanations, to enhance the quality of the data for model training and evaluation, keeping non-hate-offensive tweets with the label '0'.

4.3. ToxiGen

In this study, we analyze implicit and adversarial hate speech detection using the ToxiGen dataset [9], comprising over 274,000 text about 13 minority groups. This dataset uniquely includes a range of toxic and benign language facilitating the representation of subtle and detection evasive hate speech.

ToxiGen entries consist of statements and corresponding toxicity labels (toxic or benign). The dataset's high dimensionality stems from transforming these linguistic structures. The primary motivation for incorporating this dataset was to enhance our model with the capability to detect adversarial hate speech, and the effectiveness of the approach using a dataset known for its challenging detection characteristics.

Preprocessing: Our preprocessing for ToxiGen involved removing duplicates, correcting formatting, and addressing biases, stopwords, punctuation. The label varied from 1 to 5 in 0.3 increments, which were set to '0' if less than 3, '1; otherwise. This ensured model predicts the toxicity of each statement as a binary output, represented in a single-dimensional vector.

This research aims to develop a model adept at detecting implicit and adversarial hate speech, leveraging ToxiGen's unique properties and preprocessing strategies. Our work contributes to ongoing efforts in combating online toxicity and promoting inclusive digital spaces.

5. Methodology

Once the data underwent preprocessing, to ensure consistency across datasets, we experimented with three pretrained language models (LLMs) of varying sizes and multilinguality: BERT Base Uncased, DistilBERT, and TinyBERT. These models were selected to analyze tradeoffs between performance and computational requirements.

5.1. Pretraining

Each LLM underwent pretraining using masked language modeling (MLM) [7] on a large general text corpus. 40% of input tokens were randomly masked, and models were trained to predict masked words based on context [6].

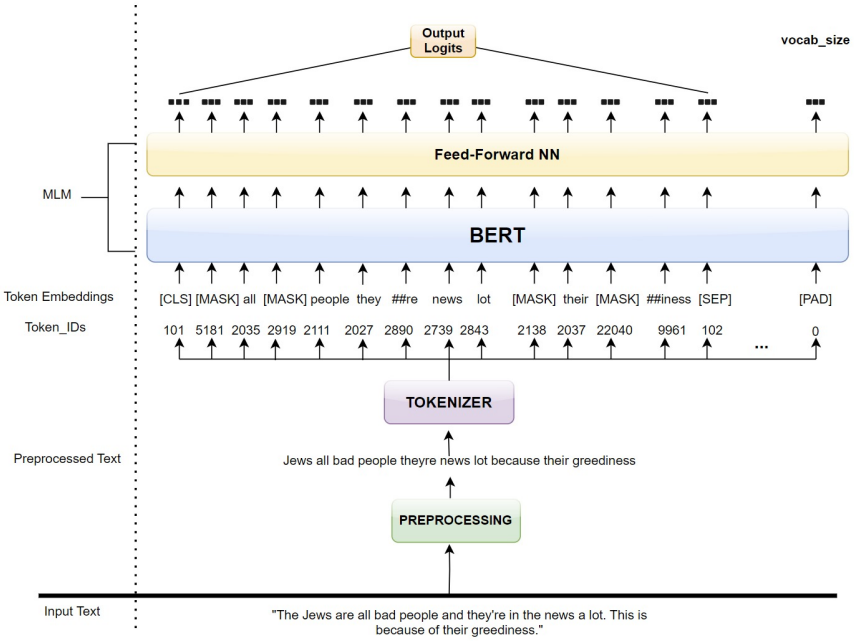

Figure 1. Masked Language Modelling in action

Initially, 15% of the data was masked, standard followed by BERT but better results were observed when the masking was raised to 40% [1]. These observations were in line with the results in [29].This enabled the models to develop capabilities such as structural and semantic understanding of language. Initially, preprocessed text-only data was extracted from datasets, tokenized where [CLS], [SEP] and [PAD] tokens were added. Randomly, 40% of data were chosen and masked (assigned 103 token value). Bert was trained to predict these masked outputs and post obtaining an optimized model, the pretrained model was obtained and pushed to a hugging face.

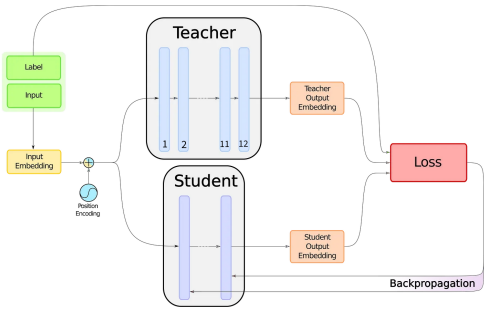

Figure 2. Custom loss funtion showcasing DistilBERT's distillation process

5.2. Knowledge Distillation

Knowledge distillation [4] is the process where a smaller model (student) is trained to replicate the behavior of a larger, more complex model (teacher). It's a technique used to compress the information contained in a large model into a smaller one. We applied knowledge distillation to transfer knowledge from the computationally expensive BERT teacher model to the smaller DistilBERT and TinyBERT student models. The teacher model's soft target probability outputs were used to provide additional supervision signals for the student models during training.

To this end [5], we use a specialized training class called DistillationTrainer. Here, along with the student and teacher models, we initialize a data collator, a tokenizer and a function to compute metrics (accuracy). The training process involves a modified loss function that combines a task-specific loss with a distillation loss. This combination of losses ensures that the student model not only learns to perform the task well but also mimics the teacher model's behavior. After training, the model's performance is evaluated using the evaluation dataset, and the trained model, along with its metrics and a model card, is then saved and can be pushed to the Hugging Face Hub. This makes the model accessible for future use.

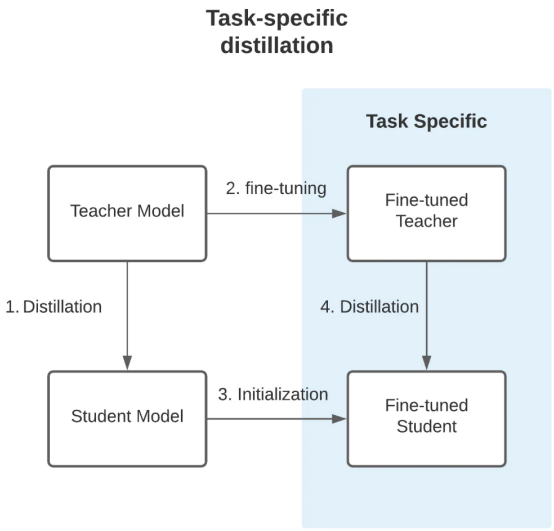

Figure 3. In Task-specific knowledge distillation a "second step of distillation" is used to "fine-tune" the model on a given dataset.

5.3. Finetuning

The pretrained and distilled models were then finetuned [10] on the labeled hate speech examples. This stage focused on adapting the general language knowledge to the specific task of hate speech classification. Models were trained to optimize metrics like accuracy, precision, recall and F1 score. We select a range of hyperparameters for tuning such as learning-rate, number of epochs, alpha and temperature and run it for a total of 50 trials to find the best combination of hyperparameters to get the best accuracy. The models are trained on our task-specific dataset containing hate-speech which are expanded further in the Data section of this paper.

The multi-stage approach of diverse pretraining data followed by finetuning on hate speech data was designed to improve model robustness, generalizability, feature representation, and training efficiency over alternative methods. We performed fine-tuning on both the parent and child models acquired through distillation to achieve the best model. Furthermore, distillation improved efficiency without sacrificing too much performance.

5.4. Justification of Approach

Our multi-dataset approach combined with extensive pretraining leveraging MLM and strategic finetuning offers several advantages over alternative methods:

1. Improved Generalizability: The model is exposed to diverse data, mitigating overfitting to any single dataset or language style. This promotes robust performance across various real-world contexts. Additionally, combining datasets helps mitigate biases that may be present in individual datasets.

2. Enhanced Feature Representation: MLM equips models to learn nuanced textual representations that capture subtleties.

3. Multiple distillation model support: In the process of distillation, the child and parent models must employ the same tokenizer. Pretraining enabled us to tokenize the parent using the designated tokenizer, thereby extending the capability.

We explored various approaches to hate speech detection, starting with rule-based systems for their speed but found them inflexible. Standard machine learning and deep learning models showed promise in learning complex patterns, yet struggled with contextual nuances and implicit hate speech. Although ensemble methods were promising, their computational cost was prohibitive. Multi-modal approaches proved impractical due to additional data requirements and increased complexity. Ultimately, we chose large language models for their efficiency, scalability, and ability to capture complex linguistic features and nuances compared to other alternatives explored.

6. Experiments and Results

The experiments in this work focused on exploring different knowledge distillation configurations with various teacher and student models to optimize hate speech detection performance. The configurations were evaluated using metrics like F1 score, accuracy, precision, recall, model size and prediction time. Comparative analysis was done against benchmark models. Extensive experiments across multiple datasets assessed model robustness and versatility.

6.1. Experimentation across Models on Hate Speech 18

We started the experimentation, with pretraining models with various complexity - Bert, TinyBert and DistilBert on the dataset `hate_speech18`[27]. These models were later finetuned and distilled on both for comprehensive comparison. The purpose was to tailor the models to the nuances and complexities inherent in hate speech classification.

To gauge the effectiveness of our approach, we explored three distinct knowledge distillation configurations. Each configuration involved a unique combination of teacher and student models, allowing us to investigate the impact of different architectures on hate speech detection.

6.1.1 BERT-base-uncased as Teacher, TinyBERT as Student

We initiated the experiments with BERT-base-uncased as the teacher model and TinyBERT as the student model. The objective was to evaluate the knowledge transfer from a

Table 1. Comparison of Different Distillation Configurations on hate_speech18 dataset.

Distillation Configuration	Model Size (params)	F1 Score	Accuracy	Precision	Recall	Prediction Time (s)
Non-Pretrained - BERT-TinyBert	4.39M	0.517	0.669	0.955	0.355	3.006
Pretrained - BERT-TinyBert	4.39M	0.778	0.812	0.951	0.659	4.58
Pretrained - BERT-DistillBERT	67M	0.895	0.904	0.989	0.818	77.458
BERT-base-uncased (Benchmark)	110M	0.667	0.501	0.5	1	135.567
facebook-roberta-hate-speech (Benchmark)	125M	0.859	0.868	0.917	0.809	134.227
BERT based hatexplain (Benchmark)	110M	0.489	0.647	0.887	0.338	134.153
DistillRoBERTa (Benchmark)	83M	0.527	0.616	0.687	0.428	68.104
distilbert-base-uncased (Benchmark)	67M	0.6535	0.5872	0.5631	0.7786	71.4281
google/bert_uncased (Benchmark)	4.43M	0.6378	0.5375	0.5242	0.8143	11.5779

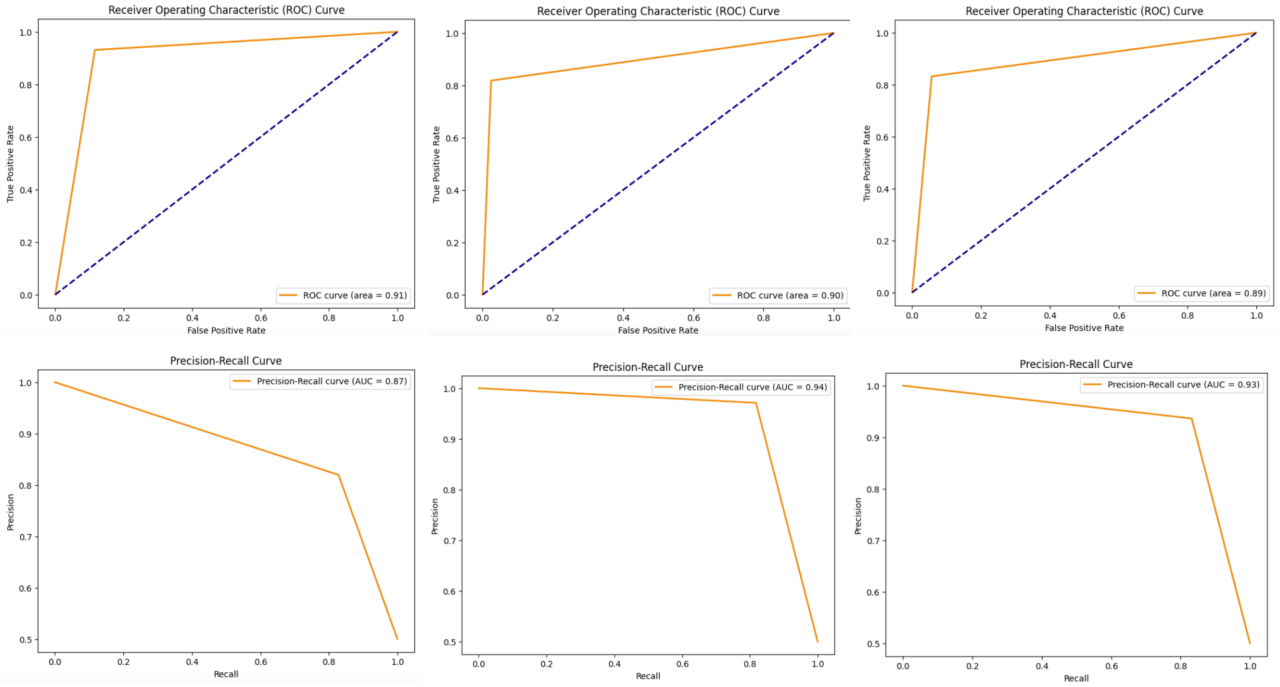

Figure 4. From left to right: ROC and PR curves of the distilled model trained on all three datasets tested on `HateXplain`; distilled model trained on all three datasets tested on `hate_speech18`; and distilled model trained on all three datasets tested on `ToxiGen`.

larger teacher model to a smaller student model. Here, the parent and student model were not pretrained but fine-tuned on the Hate Speech dataset. The results, as summarized in Table 1, indicate a moderate trade-off between accuracy and precision.

6.1.2 Pretrained BERT as Teacher, Fine-Tuned TinyBERT as Student

Recognizing the importance, we experimented with pretraining both the teacher and student models. The results demonstrate a notable improvement in performance over the first configuration, emphasizing the significance of pretraining and fine-tuning for effective knowledge distillation which can be seen from Table 1.

6.1.3 Pretrained BERT as Teacher, DistillBERT as Student

To assess the influence of a more intricate student model, we opted for DistilBERT as the child in this setup. By utilizing a pretrained and fine-tuned BERT model as the teacher, the distillation process resulted in significant enhancements in hate speech detection metrics. Despite the superior performance of this model, the second approach demonstrated greater efficiency. Nonetheless, the model managed to sur-

Table 2. Comparison of Proposed Model across datasets with Benchmarks

Configuration	Model Size (params)	F1 Score	Accuracy	Precision	Recall	Prediction Time(s)
HateSpeech18 Model	4.39M	0.778	0.812	0.951	0.659	4.58
Benchmark Model on hate_speech18	110M	0.699	0.732	0.796	0.623	144.308
HateXplain Model	4.39M	0.868	0.867	0.86	0.875	5.984
Benchmark Model on HateXplain)	110M	0.614	0.706	0.893	0.467	226.486
ToxiGen Model	4.39M	0.824	0.823	0.82	0.828	3.871
Benchmark Model on ToxiGen	110M	0.533	0.631	0.725	0.421	129.763

Table 3. Comparison of Model on aggregated dataset with other models: Generalization Depection

Model Name	Metric	hate_speech18	toxigen	hatexplain	ethos_binary (unseen dataset)	Average
Model Trained on hate_speech18	F1 score	0.778	0.73	0.7	0.52	0.682
	Accuracy	0.812	0.71	0.73	0.64	0.723
	Time	4.58	2.93	2.97	2.45	3.2325
Model trained on Toxigen	F1 score	0.34	0.824	0.4	0.69	0.5635
	Accuracy	0.577	0.823	0.59	0.61	0.65
	Time	6.38	5.984	5.64	0.85	4.7135
Model trained on HateXplain	F1 score	0.48	0.655	0.868	0.62	0.65575
	Accuracy	0.57	0.52	0.867	0.66	0.65425
	Time	9.79	4.32	5.984	1.43	5.381
Model trained on aggregated dataset	F1 score	0.887	0.88	0.9	0.72	0.84675
	Accuracy	0.89	0.887	0.9	0.75	0.85675
	Time	8.76	7.69	11.6	2.83	7.72

pass baseline models by a considerable margin in terms of both performance and efficiency.

6.1.4 Evaluation Metrics and Comparative Analysis

We opted for a diverse set of baseline models, encompassing Bert-base-uncased, distilbert, tinybert (google/bert), and benchmark models like HateXplain, DistilRoberta, and facebook-roberta-hateSpeech from Hugging Face. This selection enabled a comprehensive exploration of the proposed method, allowing for nuanced comparisons across a spectrum of performance and computational efficiency metrics.

Inference: After comprehensive analysis, we selected the Pre-trained/Fine-tuned BERT-TinyBERT as Teacher-Student configuration for further experimentations. The model demonstrates superior performance on the hate_speech18 dataset compared to baseline models with size similar to non-pretrained model but increase in accuracy (21.3%) and F1(50.4%) substantiating pre-training importance. Even though pre-trained Distilbert results in best performance, with increase in F1 (4.2%), accuracy (4.14%) and better efficiency i.e fewer parameters (45%) and faster speed (1.75x) than facebook/roberta hate speech model (model with best performance in baseline models), the HateSpeech18 model's smaller size is a critical factor for practical deployment, aligning with the efficiency and resource constraints of real-world applications.

6.2. Experimentation across Datasets on Pre-Trained Bert/TinyBert

For the next set of experimentation, we conducted extensive experiments on three distinct datasets—Hate Speech18, HateXplain, and Toxigen. The overarching goal was to assess the robustness and versatility of different model configurations, comparing their performance against established benchmarks. Initially the Bert and Tiny Bert were pre-trained on the respective datasets. These models were fine-tuned and distilled to get the final model. Additionally, we compared the models to the benchmark implementations available on Hugging Face as mentioned in Table 2.

Inference

- For Hate_Speech18 we use HateBert `https://huggingface.co/sara-nabhani/hateBert-finetuned-4` from hugging face as the benchmark model. With an increase in F1 score (11.35%), accuracy (11.03%), size reduction (25x), and reduction in prediction time (31.5x), our selected model proves to be more effective for hate speech detection in this specific context.

- For HateXplain, we used a model proposed by HateXplain research paper `https://huggingface.co/Hate-speech-CNERG/bert-base-uncased-hatexplain`. It outperformed the benchmark model, showcasing significantly higher scores with an increase in F1 Score (41.39%), Accuracy (22.79%), model size

reduction (25x), and a decrease in prediction time (37.85x). This highlights the effectiveness of our selected configuration in diverse contexts.

- For Toxigen, fine-tuned hatebert on toxigen available on Hugging Face `https://huggingface.co/tomh/toxigen_hatebert` was used. With higher F1 Score (54.65%), Accuracy (30.50%), model size reduction (25x), and prediction time reduction (33.5x), our selected approach showcases adaptability across different datasets including adversarial instances for effective hate speech detection.

6.3. Experimentation for Datasets Generalization on Pre-Trained Bert/TinyBert

In the concluding phase of our experiments, our objective was to propose a model that achieves robust generalization through the amalgamation of diverse dataset sources. To ensure this capability, we amalgamated all datasets into a unified text document, leveraging Bert and Tiny Bert for pretraining, followed by their deployment on Hugging Face. Subsequently, these models underwent fine-tuning on the combined dataset from all three sources and were distilled to yield the final proposed model.

6.3.1 Evaluation Metrics and Comparative Analysis

A thorough analysis was conducted for each dataset-specific model, extracting cross-data metrics. The same methodology was applied to the aggregated model to gauge its performance. Additionally, all models underwent testing on an entirely unfamiliar dataset - Ethos binary classification `https://huggingface.co/datasets/ethos`. The resulting metrics were then compared, and averages were computed to assess the generalization offered by the proposed model.

Inference: The aggregated model performed exceptionally well on the datasets. It was able to generalize well enough. When compared to hate_speech18 dataset model, it was able to improve F1 score (12.28%) and accuracy (8.76%). For the HateXplain dataset, F1 score (3.55%) and accuracy (3.66%) saw a slight increase. Finally, for the toxigen dataset the combined model was able to showcase moderate enhancement of F1 score (6.39%) and accuracy (7.21%). The model also had the best prediction on an unseen dataset, Ethos. In conclusion, compared to other models, the aggregated model exhibited the highest average F1 score and accuracy. It asserts our proposal of having better generalizability of a model trained on aggregated datasets.

7. Limitations

Our hate speech detection model demonstrates robust performance but has limitations. It relies on specific hate speech datasets, cautioning against broader application due to reduced Ethos dataset performance. Computational constraints limit exploration, impacting training epochs and hyperparameters. Biases may inadvertently transfer from training data, necessitating ongoing identification and mitigation efforts. Primarily text-focused, the model encounters challenges with multimedia content. Dynamic online platforms demand continuous adaptation, influencing the model's performance. Recognizing these limitations is crucial for responsible use, guiding future iterations to enhance adaptability and fairness.

The hate speech detection model performs robustly yet faces challenges. Its efficacy hinges on specific datasets, cautioning against broader application due to reduced Ethos dataset performance. Computational constraints limit exploration, impacting training epochs and hyperparameters. Biases may transfer from training data, necessitating ongoing identification and mitigation efforts. Primarily text-focused, the model encounters challenges with multimedia content. Dynamic online platforms demand continuous adaptation, influencing the model's performance. Recognizing these limitations is crucial for responsible use, guiding future iterations to enhance adaptability and fairness.

8. Conclusion

Our hate speech detection model, featuring pretrained, fine-tuned BERT/TinyBERT, demonstrates exceptional efficacy in addressing online hate speech challenges. The optimal model configuration strikes a judicious balance between size, computational efficiency, and robust hate speech detection. Across diverse datasets, our model consistently outperforms benchmarks, showcasing its versatility and reliability. Trade-off considerations emphasize the advantage of prioritizing a smaller model size, facilitating practical deployment in resource-constrained environments. The consolidated method excels in generalization, comprehending and adapting to diverse data, outperforming specific models even on unseen data.

Looking forward, the future scope involves exploring multimodal integration and dynamic adaptability, aiming to enhance the model's capabilities. Additionally, we propose integrating efficient models with practical applications, bridging the gap between research and real-world impact. The exploration of ensembling techniques for performance enhancement presents a promising avenue for future research. In essence, our findings contribute to safer online environments and responsible digital discourse. As we progress, these insights serve as a catalyst for ongoing advancements, fostering a more inclusive and secure online space.

References

[1] Build a masked language model with transformers. https://github.com/ayoolaolafenwa/TrainNLP.

[2] Hate speech 18 dataset. https://huggingface.co/datasets/hate_speech18.

[3] Hatexplain dataset. https://huggingface.co/datasets/hatexplain.

[4] Knowledge distillation neptune. https://neptune.ai/blog/knowledge-distillation.

[5] Knowledge distillation simple tutorial. https://towardsdatascience.com/simple-tutorial-for-distilling-bert-99883894e90a.

[6] Learn transformers. https://github.com/jamescalam/transformers.

[7] Masked language modeling with hugging face transformers: A beginner's guide. https://medium.com/@lokaregns/masked-language-modeling-with-hugging-face-transformers-a-beginners-guide-b50af74a7560.

[8] Preliminary analysis on china-related hateful tweets. https://github.com/JINHXu/how-much-hate-with-china.

[9] Toxigen dataset. https://huggingface.co/datasets/skg/toxigen-data.

[10] Ultimate guide to llm finetuning. https://www.lakera.ai/blog/llm-fine-tuning-guide.

[11] F. Alkomah, S. Salati, and X. Ma. A new hate speech detection system based on textual and psychological features. *International Journal of Advanced Computer Science and Applications*, 13(8), 2022.

[12] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*, 2020.

[13] T. A. Belal, G. Shahariar, and M. H. Kabir. Interpretable multi labeled bengali toxic comments classification using deep learning. In *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. IEEE, 2023.

[14] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*, 2020.

[15] L. Gao and R. Huang. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*, 2017.

[16] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

[17] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[18] J. Kim, B. Lee, and K.-A. Sohn. Why is it hate speech? masked rationale prediction for explainable hate speech detection. *arXiv preprint arXiv:2211.00243*, 2022.

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

[20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[21] A. C. Mazari, N. Boudoukhani, and A. Djeffal. Bert-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*, pages 1–15, 2023.

[22] J. Melton, A. Bagavathi, and S. Krishnan. Del-hate: deep learning tunable ensemble for hate speech detection. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1015–1022. IEEE, 2020.

[23] A. Nayla, C. Setianingsih, and B. Dirgantoro. Hate speech detection on twitter using bert algorithm. In *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, pages 644–649. IEEE, 2023.

[24] G. Rajput, N. S. Punn, S. K. Sonbhadra, and S. Agarwal. Hate speech detection using static bert embeddings. In *Big Data Analytics: 9th International Conference, BDA 2021, Virtual Event, December 15-18, 2021, Proceedings 9*, pages 67–77. Springer, 2021.

[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[26] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020.

[27] J. Thomas. Code used for knowledge distillation. https://github.com/josephgit10/knowledge-distillation-bert.

[28] B. Vidgen, T. Thrush, Z. Waseem, and D. Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*, 2020.

[29] A. Wettig, T. Gao, Z. Zhong, and D. Chen. Should you mask 15

[30] S. Zimmerman, U. Kruschwitz, and C. Fox. Improving hate speech detection with deep learning ensembles. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, 2018.

Supplementary Material

1. Hyperparameter Optimization

The initial step involved defining a hyperparameter search space (hp_space function). Key hyperparameters included the number of training epochs (num_train_epochs), the learning rate (learning_rate), a distillation-specific alpha parameter (alpha), and the temperature parameter (temperature). These hyperparameters were selected for optimization due to their significant impact on the training dynamics and final model performance.

The learning rate was explored in a logarithmic scale between 1×10^{-6} and 1×10^{-4}, a range typically effective for fine-tuning tasks in natural language processing. The alpha parameter, crucial in balancing the contributions of teacher model outputs and ground truth labels during distillation, was varied between 0 and 1. The temperature parameter, impacting the softening of probabilities in distillation, was explored between 2 and 30.

1.1. Model Initialization and Distillation Process

The student model was initialized using the AutoModelForSequenceClassification class, aligning with the task-specific requirements. This initialization process incorporated necessary configurations such as the number of labels and their corresponding mappings.

A DistillationTrainer object managed the distillation process. It encapsulated the student model initialization, training and evaluation datasets, data collation procedures, tokenization methods, and metric computation functions.

1.2. Hyperparameter Search and Model Training

The objective was to maximize a specified performance metric over a total of 10 trials. Upon identifying the optimal hyperparameters, these were applied to update the training arguments of the student model.

Subsequently, the student model underwent training with the optimized parameters using a new DistillationTrainer instance. This process ensured that the distilled model was fine-tuned under conditions conducive to achieving high performance.

1.3. Model Saving and Distribution

Upon completion of the training phase, the model, along with its performance metrics, was saved. Additionally, a model card detailing the model's specifications and training settings was created. The final model, along with its associated metrics and documentation, was then pushed to a model hub for accessibility and utilization by the wider research community.

2. Future Directions

This study lays the groundwork for several promising avenues of research in the domain of hate speech detection, each with the potential to significantly advance the field:

- **Multimodal Data Integration**: Future studies should explore the incorporation of multimodal data sources, such as images and videos, into hate speech detection frameworks. This approach will enable a more comprehensive analysis of content, capturing nuanced expressions of hate speech in diverse media formats

- **Adaptability to Evolving Communication Patterns**: Given the dynamic nature of online communication, it is imperative to develop models capable of adapting to new patterns and forms of hate speech. Continuous learning mechanisms may be integrated to ensure the model's ongoing relevance and efficacy.

- **Cross-Dataset Generalization**: Enhancing the model's ability to generalize across diverse datasets is critical. Future work should focus on improving performance across various linguistic and cultural contexts, thereby broadening the model's applicability.

- **Bias Mitigation and Ethical Considerations**: Ongoing efforts must be directed towards identifying and mitigating biases within hate speech detection models. Ethical considerations should be at the forefront, ensuring fairness and minimizing potential harm.

- **Interpretability and Explainability**: Enhancing the model's interpretability and explainability is a key area for future research. This will not only foster trust in the model's decisions but also provide valuable insights into the complex dynamics of hate speech.

- **Resource Efficiency and Deployment**: Research should continue to focus on optimizing models for

resource efficiency, particularly for deployment in resource-constrained environments. This includes exploring advanced model compression techniques and efficient architectures.

- **Real-time Processing and Deployment**: Optimizing models for real-time detection on various platforms represents a significant research opportunity. This involves improving processing speed and efficiency for immediate detection and action.
- **Collaborative and Interdisciplinary Research**: Engaging in collaborative research, involving disciplines such as social sciences, ethics, and law, is crucial. Such interdisciplinary approaches can offer a more comprehensive understanding of hate speech and contribute to the development of well-rounded solutions.

By pursuing these directions, future research can significantly contribute to the development of advanced, responsible, and effective hate speech detection mechanisms, thereby enhancing online discourse and promoting digital safety.

2.1. Data Preprocessing

The preprocessing pipeline was essential for preparing the text data for input into the BERT-based models. This pipeline encompassed several key steps:

- **Tokenizer Initialization:**

  We initialized two instances of AutoTokenizer from the Hugging Face Transformers library, both utilizing the pre-trained models. One tokenizer was designated for the teacher model (teacher_tokenizer) and the other for the student model (student_tokenizer). This ensured consistency in the way text data was tokenized, aligning with the pre-training schema of the BERT models.

- **Text Processing Function:**

  A custom function, process, was defined to tokenize the text data. It processed the "text" field of the input examples, applying tokenization with truncation to maintain a uniform sequence length, and setting a maximum length of 256 tokens. This standardization was crucial for maintaining consistency across the dataset.

- **Application of Tokenization:**

  The process function was systematically applied across the dataset splits (e.g., training and validation) using a map function. Subsequent to tokenization, the label column in the dataset was renamed to "labels" and any unnecessary columns were dropped, preparing the data for the sequence classification task.

- **Label Encoding:**

  To facilitate the classification task, we established a mapping between labels and numerical identifiers. Two dictionaries, label2id and id2label, were created for this purpose. This step ensured that categorical labels were appropriately transformed into a numerical format, interpretable by the models.

- **Data Collator Initialization:**

  A DataCollatorWithPadding was initialized using the tokenizer. This collator dynamically padded the input sequences to uniform lengths during training. This approach optimized batch processing, ensuring efficient handling of varying sequence lengths.

3. Results and Evaluation:

For the models used in the paper, we evaluated the model by using metrics like accuracy, precision, recall, f1-recall, Precision-Recall Curve, ROC curve and Token-level attention weights.

Classification Report:

	precision	recall	f1-score	support
0	0.74	0.97	0.84	533
1	0.95	0.66	0.78	533
accuracy			0.81	1066
macro avg	0.85	0.81	0.81	1066
weighted avg	0.85	0.81	0.81	1066

Figure 1. Metrics for BERT/TinyBERT Model pretrained on hate_speech18

More results were obtained for different models and the evaluation results are mentioned in the attached document: https://docs.google.com/document/d/1t_Ob95IFekqGQRI-jeSRhfGakFQiZ2fhy1_BKnN9lqU/

All the models developed and compared can be accessed via the result document. Additionally, they were made publically available and can be accessed on the following profiles - datasets and preprocessed models: https://huggingface.co/agvidit1. Finetuned and distilled models: https://huggingface.co/joseph10.

3.1. Weights and tokens visualization

We also visualized the weights and tokens of some of the models we used to understand the model better.

The weights and tokens visualization of other few models is mentioned in this document: https://docs.google.com/document/d/1ttva0H-Pd8MEW9KkEhIXrP5oiNdlTWkDRUli__gxJa4/

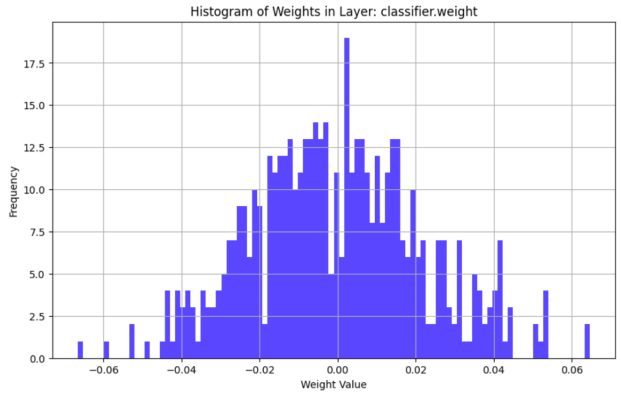

Figure 2. Layer weights of pretrained distilled model on hate_speech18

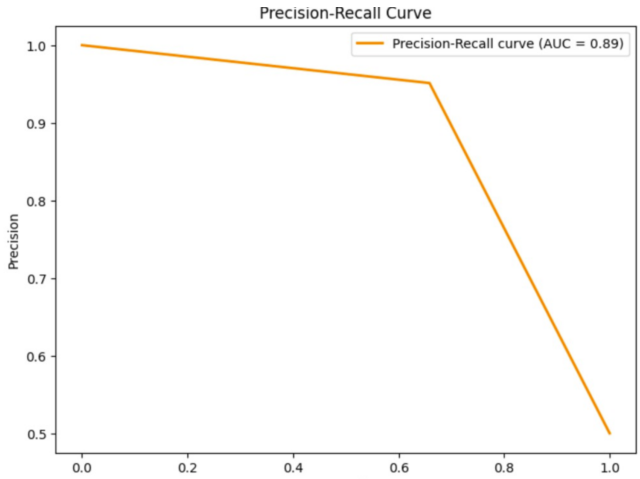

Figure 3. Precision-Recall Curve of BERT/TinyBERT Model pretrained on hate_speech18

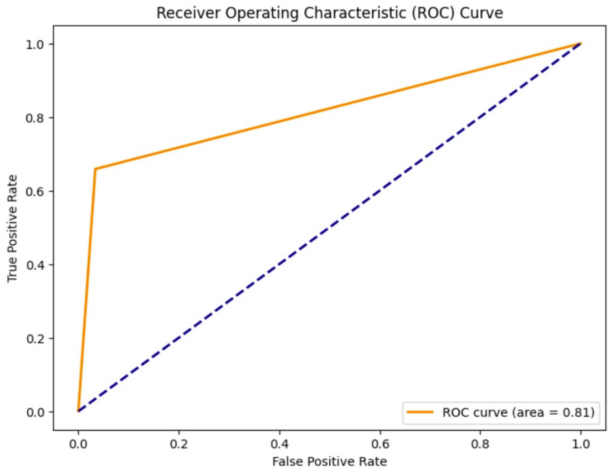

Figure 4. ROC Curve of BERT/TinyBERT Model pretrained on hate_speech18

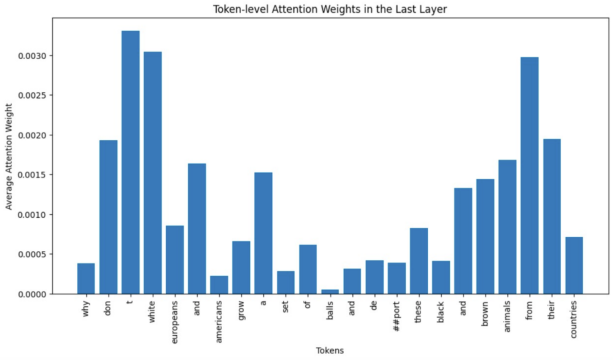

Figure 5. Average attention weights vs Token-level attention weight in last layer of BERT/TinyBERT Model pretrained on hate_speech18