Toxicity/Hate Speech Classification

Ravi Raghavan **Dhruv Verma** **Raafae Zaki** **Ashwin Balaji**

Abstract

In this work, we examine hate speech detection on social media using MetaHate, a large-scale collection of 36 datasets comprising over 1.1 million posts. We begin with a majority-class baseline to establish a minimal performance benchmark, then fine-tune a BERT model as a contextual text classifier. To enhance performance, we explore architectural extensions including BERT+CNN, which captures local n-gram patterns, and BERT+BiLSTM, which models sequential dependencies. We further combine models through ensemble methods: hard voting, soft voting, max voting, and stacking. Our best-performing hard voting ensemble achieves a test F1 of 0.726, demonstrating improved contextual understanding and overall robustness in hate speech detection.

1 Introduction

Hate speech detection involves automatically identifying harmful or offensive content directed at individuals or groups in social media posts. This task is central to natural language processing (NLP) and computational linguistics because it requires understanding subtle linguistic cues, context, and the pragmatic meaning of text. Detecting hate speech goes beyond keyword matching, as phrases may appear harmless in isolation but be hateful in context. Contextual language models like BERT are particularly effective, as they can capture semantic, syntactic, and pragmatic information from text.

With the rise of social media, automated detection is crucial for content moderation, public safety, and research on online behavior. This task intersects with computational linguistics challenges such as detecting sarcasm, figurative language, or implicit hate, requiring models to interpret both lexical patterns and broader discourse context.

An illustrative example of this task is shown in Table 1, where each post must be classified as *Hate Speech* or *Non-Hate Speech*. Subtle differences in phrasing, tone, or target can change the label: the first post uses explicit negative language and attacks an individual, clearly constituting hate speech; the second expresses frustration without targeting anyone, labeled non-hate speech. This highlights the challenge of detecting not only overtly offensive language but also context, implied meaning, and social intent.

Post	**Label**
I hate [Group X], they are the worst.	Hate Speech
I can't believe people still say this online	Non-Hate Speech

Table 1: Illustrative examples of posts and their corresponding labels. The task is to predict the correct label given the text of a social media post.

Formally, the problem can be defined as follows. Let $\mathcal{X}$ denote the space of social media posts and $\mathcal{Y} = \{0, 1\}$ denote the label space, where 0 represents non-hate speech and 1 represents hate speech. The goal is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps each post $x \in \mathcal{X}$ to its correct label $y \in \mathcal{Y}$. The function f is typically learned from a labeled dataset $\{(x_i, y_i)\}_{i=1}^{N}$ using supervised machine learning techniques, with the objective of minimizing a classification loss (e.g., cross-entropy) and maximizing metrics such as F1 score, especially for the hate speech class.

We selected this task for our term project because of its societal relevance. Automated hate speech detection is crucial for fostering safer online communities and protecting vulnerable groups from harassment and discrimination. The task also highlights the broader societal challenge of moderating content in ways that balance safety with freedom of expression. Deep learning techniques, particularly transformer-based architectures, offer significant value in this context by capturing subtle contextual cues, nuanced language, and implied meaning that simpler models often miss. By leveraging these methods, we can build systems that more effectively detect harmful content, providing practical tools for mitigating online abuse while advancing research in socially impactful NLP applications.

2 Literature Review

2.1 A Comprehensive Review on Automatic Hate Speech Detection in the Age of the Transformer

This survey paper (Ramos et al., 2024) presents a systematic and up-to-date review of the evolution of automatic hate speech detection methods, with a particular emphasis on the paradigm shift introduced by Transformer-based architectures. Following PRISMA guidelines, the authors analyze over 100 peer-reviewed studies published since the introduction of Transformers, tracing the progression from traditional machine learning approaches (e.g., SVMs and logistic regression) and deep learning models (CNNs, LSTMs, GRUs) to modern transfer-learning-based architectures such as BERT, RoBERTa, XLM-RoBERTa, ELECTRA, and language-specific variants. The review provides a structured comparison across five methodological categories, namely traditional machine learning, deep learning, Transformer models, generative models, and multi-task learning, and evaluates their performance, data requirements, and computational trade-offs. Across nearly all benchmark settings, Transformer-based models consistently achieve state-of-the-art results, particularly in multilingual and cross-lingual scenarios, though at the cost of higher computational complexity. The authors also synthesize trends in dataset usage, highlighting a strong dominance of English-language Twitter data alongside growing but still limited work on low-resource languages and alternative platforms. Emerging directions such as generative data augmentation and multi-task learning with emotion or sentiment supervision are identified as promising but underexplored. Importantly, the survey emphasizes persistent challenges including dataset fragmentation, lack of standardized benchmarks, algorithmic bias, and limited reproducibility due to unavailable code and data. Overall, the paper positions Transformer-based models as the current cornerstone of hate speech detection research while arguing for hybrid, fair, and resource-aware solutions to improve generalization, transparency, and inclusivity in future systems.

Building on this literature, we adopt the shared-task paradigm commonly used in hate speech detection by framing our work as a standardized benchmarking exercise. Specifically, we fine-tune BERT on the MetaHate dataset and extend it with CNN and LSTM architectures, evaluating all models on MetaHate to directly compare performance and assess the benefits of hybrid transformer–neural approaches under a unified dataset and evaluation setting.

2.2 MetaHate: A Dataset for Unifying Efforts on Hate Speech Detection

This paper (Piot et al., 2024) introduces MetaHate, a large-scale meta-collection designed to address fragmentation and inconsistency across existing hate speech datasets. Rather than proposing a new model architecture, the authors focus on dataset unification, systematically reviewing over 60 publicly available hate and abusive speech corpora and integrating 36 English-language social media datasets into a single benchmark. The resulting collection contains over 1.22 million de-duplicated posts drawn from diverse platforms including Twitter, Reddit, Gab, Stormfront, YouTube, Wikipedia, and Facebook, unified under a binary hate versus non-hate labeling scheme aligned with United Nations definitions. The authors carefully filter datasets to ensure conceptual consistency, excluding synthetic data and corpora whose definitions conflate hate with general offensiveness. In addition to dataset construction, the paper provides extensive corpus analysis, including lexical frequency analysis, named-entity distributions, topic modeling with LDA, psycholinguistic emotion analysis using Plutchik's framework, and t-SNE visualizations, revealing systematic differences between hate and non-hate language in terms of targets, emotional tone, and linguistic structure. To establish baseline performance, the authors evaluate traditional and neural models, including an SVM with TF-IDF features, a CNN, and BERT, showing that BERT achieves the strongest performance with an accuracy of 0.89 and a macro-F1 of 0.80, highlighting the advantages of pretrained language models on large, heterogeneous hate speech data. Overall, the paper positions MetaHate as the first large, unified benchmark for hate speech detection, emphasizing that dataset scale, conceptual clarity, and cross-source diversity are critical for training robust and generalizable detection models, while noting limitations related to binary labeling, English-only scope, and the need for contextual and multilingual extensions in future work.

2.3 BERT-based Ensemble Approaches for Hate Speech Detection

This paper (Mnassri et al., 2022) investigates how combining transformer-based models and neural networks can improve the automated detection of hate speech on social media platforms such as Twitter. The authors extend the capabilities of BERT (Bidirectional Encoder Representations from Transformers) by integrating it with three different neural architectures, a Multi-Layer Perceptron (MLP), a Convolutional Neural Network (CNN), and a Long Short-Term Memory (LSTM) network, to capture complementary linguistic features. Using three publicly available Twitter datasets (Davidson, HatEval2019, and OLID), they created a unified, more balanced dataset called DHO, designed for multi-label classification of hateful, offensive, and neutral content. To enhance model performance and robustness, the study employed four ensemble learning methods: (1) Soft voting, where each model outputs class probabilities and the final prediction is the class with the highest average probability across models. (2) Max voting, which selects the class predicted with the highest individual probability among all models. (3) Hard voting, which uses the majority vote among the models' discrete predictions (e.g., if two out of three predict "offensive," that becomes the final label). (4) Stacking (stacked generalization), a more advanced technique where predictions from base models (e.g., BERT+MLP, BERT+CNN, BERT+LSTM) are fed into a meta-classifier, in this case, a linear regression model, that learns to optimally combine them for the final output. The experiments showed that, for the most part, BERT+NN hybrid models outperformed the plain BERT baseline, highlighting the benefit of combining contextual embeddings with neural feature extractors. Ensemble techniques further improved performance: on the Davidson dataset, the stacking ensemble achieved a remarkable 97% F1-score, surpassing other ensemble methods. On the larger DHO dataset, ensembles involving BERT+MLP and BERT+LSTM achieved the best macro-F1 and precision scores, though at a higher computational cost. Overall, the paper demonstrates that combining transfer learning (BERT) with ensemble learning yields superior hate speech detection performance, capturing subtle and context-dependent language. The authors note that future work should address class imbalance, computational efficiency, and interpretability, potentially through architectures like K-BERT or enhanced ensemble framework.

2.4 HateBERT: Retraining BERT for Abusive Language Detection in English

This paper (Caselli et al., 2020) from Caselli et al., 2021 introduces HateBERT, a version of BERT specially retrained to improve the detection of abusive and hateful language on social media. The authors aim to make BERT more sensitive to toxic and aggressive forms of online communication by adapting it to the domain of abusive language. To do this, they used domain-specific fine-tuning, where the standard English BERT model (base uncased) is further trained on language drawn from online communities known for toxic behavior. The new training data, called RAL-E, contains over one million Reddit comments collected from communities that were banned for promoting offensive, abusive, or hateful content. By fine-tuning the model to this kind of text, the authors tried to shift BERT's understanding of language toward the patterns and tone found in abusive speech. The team compared the retrained HateBERT model with the original BERT across three benchmark datasets that focus on different aspects of harmful online language: OffensEval 2019 for offensive language, AbusEval for abusive language, and HatEval for hate speech. In all cases, HateBERT achieved higher macro-averaged F1 scores, showing that it was better at identifying both harmful and non-harmful messages. The biggest improvement appeared on the AbusEval dataset, where HateBERT even surpassed previous best results. The authors also tested how well the models transferred between datasets. These portability experiments showed that HateBERT performed more reliably across related tasks, especially when moving from broader categories like offensive language to more specific ones like hate speech. However, they also found that the ability to generalize depends on how similar the labeled categories are between datasets. Overall, the paper shows that further fine-tuning of LLMs on domain-specific data is an effective and low-cost way to adapt them to specialized tasks such as hate speech detection. The authors released both the HateBERT model and the RAL-E dataset to support future research.

2.5 Advancing Hate Speech Detection with Transformers: Insights from the MetaHate

The paper (Chapagain et al., 2025) presents a large-scale study on transformer-based models for hate speech detection using the previously introduced MetaHate dataset. The authors begin by outlining the limitations of traditional machine learning and deep learning approaches, such as SVMs, CNNs, and LSTMs in capturing the complex contextual nuances of hate speech. They then systematically evaluate several transformer architectures, including BERT, RoBERTa, GPT2, BART, DeBERTa, Longformer, XLNet, T5, and ELECTRA, to identify which model most effectively detects hate speech across diverse and noisy social media data. The dataset was tokenized, balanced using class weights, and all models were fine-tuned under consistent hyperparameter settings to ensure fair comparison. Among all evaluated models, ELECTRA achieved the best performance with an F1 score of 0.8980 and accuracy of 0.8946, outperforming all baselines as well as other transformer architectures. ELECTRA's generator–discriminator framework, where the generator replaces masked tokens and the discriminator identifies those replacements, enables it to capture subtle contextual cues, making it especially effective at detecting implicit or coded hate speech. Error analysis revealed ongoing challenges with sarcasm, figurative language, and label noise. Overall, the study demonstrates that transformer-based architectures, particularly ELECTRA, substantially advance hate speech detection, while future work should focus on explainable and multimodal extensions to enhance robustness and fairness.

3 Experimental Design

3.1 Data

3.1.1 Dataset Description

We use **MetaHate**, a large-scale meta-collection of hate speech datasets compiled from social media platforms, for the task of binary hate speech detection. MetaHate aggregates **36 publicly available hate speech datasets** and contains a total of **1,226,202 posts**, of which **1,101,165 instances** are publicly available and used in this project. Each post is labeled as either hate speech or non-hate speech. The dataset is distributed in **TSV format** and consists of two fields: the social media post text and a binary label. Labels are defined as 0 for non-hate speech and 1 for hate speech

3.1.2 Data Examples

Table 2 presents representative examples illustrating the structure of the dataset. To avoid including graphic or profane content, the examples shown are hypothetical but reflect the format and labeling scheme of the actual data.

Text	Label
I can't believe people still say this online	0
I hate [Group X], they are the worst.	1

Table 2: Illustrative examples from the MetaHate dataset.

3.1.3 Data Splits and Statistics

Due to the large size of the full dataset, we first sampled **3% of the publicly available data** using a **stratified sampling strategy by label**. We then performed a stratified 80%/10%/10% train/development/test split to preserve the label distribution across all splits. We ensured that no post appears in more than one split. Table 3 summarizes the number of posts in each split, the average number of sentences per post across the splits, as well as the average number of words per post across the splits. We note that sentence and word count were obtained through `nltk`.

3.1.4 Label Distribution

As shown by Figure 1, the original MetaHate dataset exhibits a skewed label distribution, with non-hate speech posts occurring significantly more frequently than hate speech posts. Stratified sampling and splitting were used to ensure that this imbalance is consistently represented across the training, development, and test sets. To account for this imbalance during training, we employed a class-weighted cross-entropy loss, giving higher weight to the minority (hate speech) class.

3.1.5 Data Collection

This work uses the MetaHate dataset, published by (Piot et al., 2024). MetaHate is a large-scale, harmonized collection of 36 hate speech datasets, compiled from an initial review of over 60 datasets dedicated to hate speech detection. The dataset focuses on social media text authored by humans and excludes synthetic data or sources outside social media, such as news comments or video game

Split	Post Count	Avg. Sentences/Post	Avg. Words/Post
Train	26,427	3.13	55.3
Dev	3,303	3.11	54.9
Test	3,304	3.17	55.2

Table 3: Dataset sizes for each split.

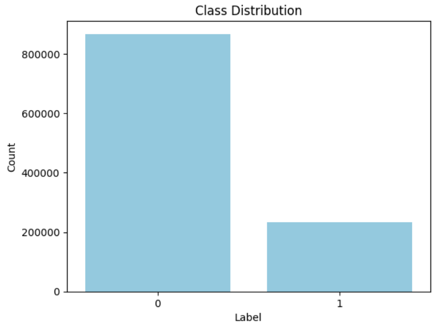

Figure 1: Class Imbalance in Original MetaHate Dataset

chats. To collect the data, the authors conducted a thorough exploration in search engines and data repositories, supplemented by a comprehensive review of academic literature. Each dataset was carefully evaluated to ensure compatibility with the MetaHate criteria, including (1) containing only human-authored social media text, (2) adhering to a defined hate speech standard (excluding general offensive content), and (3) focusing on English-language content for consistency and coverage. The resulting MetaHate dataset contained 1,667,496 posts, which were filtered down to 1,226,202 non-duplicated comments. Posts were collected from multiple social media platforms, including Twitter, Facebook, Reddit, Stormfront, Gab, Whisper, Wikipedia, Civil Comments, YouTube, and BitChute. Collection strategies included the use of lexicons, keywords, hashtags, phrase patterns, and random sampling from sources likely to contain hate speech. The dataset uses a binary classification scheme (hate vs. no hate) to broaden its applicability.

3.2 Evaluation Metric

Given the binary nature of this classification task, the F1 score for the hate speech class is employed as the primary evaluation metric, as it provides a balanced assessment of precision and recall. Recall quantifies the proportion of actual hate speech instances correctly identified, ensuring that harmful content is not overlooked, whereas precision measures the proportion of instances predicted as hate speech that are truly hateful, mitigating false positives. In the context of the MetaHate dataset, both metrics are essential: failure to detect hate speech permits the propagation of harmful content, while false positives may unjustly restrict legitimate expression. Accordingly, the F1 score, computed for the hate speech class, provides a single interpretable metric capturing this trade-off through the harmonic mean of precision and recall.

As shown in (Hossin and Sulaiman, 2015) and (Vujović, 2021), Precision, Recall, and F1 score are defined as follows: Precision $= \frac{\text{TP}}{\text{TP+FP}}$, Recall $= \frac{\text{TP}}{\text{TP+FN}}$, F1 Score $= \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$, where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively.

We review prior works that utilize the F1 score as a primary evaluation metric for hate speech detection. In particular, we see that both (Piot et al., 2024) and (Chapagain et al., 2025) leverage the MetaHate dataset and report the F1 score to assess the effectiveness of their proposed approaches in identifying hate speech.

3.3 Simple baseline

This baseline is a majority-class classifier that ignores text features and always predicts the most frequent class in the training set, which is non-hate (0). On unseen data, it predicts 0 for all examples, providing a naive benchmark that sets the minimum performance threshold for more advanced models like BERT. Evaluated on our test set, it achieved an `F1` score of 0.0, since it never predicts hate speech (1), which comprises only 20% of posts. Although it attains 80% accuracy by matching the majority class, it fails entirely at detecting harmful content, illustrating why accuracy is misleading on imbalanced datasets and why models that learn textual patterns are necessary.

4 Experimental Results

4.1 Published Baseline

As our published baseline, we implemented a fine-tuned `bert-base-uncased` model for binary hate speech detection following the methodology introduced in *MetaHate: A Dataset for Unifying Efforts on Hate Speech Detection* (Piot et al., 2024). The model leverages BERT's contextualized representations and is fine-tuned using supervised learning on the MetaHate dataset. All input posts are tokenized using BERT's WordPiece tokenizer with truncation and padding to a maximum sequence length of 512 tokens.

The MetaHate dataset is highly imbalanced, with hateful content comprising roughly 20% of the data. To address this, we employ a class-weighted cross-entropy loss, assigning each class a weight based on the frequency of the other class. This encourages the model to place greater emphasis on correctly identifying hateful posts. The model is trained for three epochs with a batch size of 32 and a learning rate of 5×10^{-5} using the AdamW optimizer. We evaluate performance using the F1 score for the hate speech class, which balances precision and recall and is well-suited for imbalanced classification tasks.

Our implementation achieves an F1 score of 0.712 on the test set, substantially outperforming a majority-class baseline. However, our results do not reach the performance reported in the original MetaHate paper. This discrepancy is expected, as the original model was trained on the full dataset of over 1.1 million samples, whereas we trained and tested on a 3% subsample due to computational constraints. Consequently, our results are not directly comparable to those reported in the paper, as both the training and test sets differ. Moreover, from a statistical learning perspective, increasing the training set size generally reduces both the bias and variance of a model; thus, as the size of the training set increases, we would expect improved generalization performance and lower population risk (i.e., lower test error). This also accounts for why (Piot et al., 2024) reported superior results.

4.2 Extensions

We explored several extensions to the baseline to assess whether architectural modifications or ensembling techniques could improve performance. First, we experimented with hybrid architectures that add sequence modeling components on top of BERT embeddings. BERT provides powerful contextual embeddings and effectively captures relationships between words across a sentence, but it has certain limitations. While BERT incorporates positional encodings to represent the sequential positions of tokens, its self-attention mechanism is inherently permutation-invariant, meaning that, irrespective of token order, it considers all token relationships equivalently, potentially yielding similar outputs. Consequently, changes in word order, such as "They are allies" versus "Allies they are," may not be fully distinguished, even though the first is neutral or positive, while the second could be interpreted sarcastically or as hostile depending on context. To address these limitations, we proposed two model extensions: BERT + CNN and BERT + BiLSTM.

The BERT+CNN model applies convolutional layers to capture local n-gram patterns indicative of hate speech, such as recurring offensive phrases. By applying convolutional layers on top of BERT embeddings, the model can detect these local patterns while still leveraging BERT's contextual understanding. This combination allows the model to be sensitive to both global context and meaningful local word combinations. While this approach improved validation performance slightly over the baseline, it resulted in a lower test F1 score (0.706), suggesting limited generalization gains.

The BERT+BiLSTM model explicitly models word order and sequential dependencies by feeding BERT embeddings into a bidirectional LSTM. This enhances sensitivity to word order and helps BERT compensate for self-attention's permutation-invariance. By using a BiLSTM on top of BERT embeddings, the model can better capture sequential dependencies, including long-range interactions between words and phrases where rearranging words changes meaning. This is especially useful for handling negations, sarcasm, or multi-clause statements often found in hate speech. This extension achieved the strongest single-model performance, with a test F1 score of 0.724, indicating improved handling of negation, long-range dependencies, and multi-clause hateful expressions.

Ensembling Methods In addition to architectural extensions, we explored multiple ensembling strategies to combine the strengths of heterogeneous models, namely the BERT baseline, BERT+CNN, and BERT+BiLSTM. Ensembling aims to reduce variance and improve robustness by aggregating predictions from models with complementary error patterns.

Hard Voting assigns each model an equal vote based on its predicted class label. The final prediction is the class that receives the majority of votes:

$$\hat{y} = \mathrm{argmax}_{c\in\{0,1\}} \sum_{m=1}^{M} \mathbb{I}(\hat{y}_m = c),$$

where $\hat{y}_m$ denotes the predicted label from model m, M is the number of models, and $\mathbb{I}(\cdot)$ is the indicator function. This approach is simple and robust to poorly calibrated probabilities.

Soft Voting aggregates predicted class probabilities instead of discrete labels. The final prediction is obtained by averaging the probabilities across models and selecting the most likely class:

$$\hat{y} = \text{argmax}_{c \in \{0,1\}} \frac{1}{M} \sum_{m=1}^{M} p_m(y = c \mid x).$$

Soft voting leverages confidence information from each model but assumes reasonably well-calibrated probability estimates.

Max Voting selects the class with the maximum confidence score across all models:

$$\hat{y} = \text{argmax}_{c \in \{0,1\}} \max_{m} p_m(y = c \mid x).$$

This strategy favors strong signals from any individual model, which can improve recall for rare but salient hate speech patterns, at the risk of increased false positives.

Stacking introduces a meta-learning approach, where a second-level classifier is trained to combine model outputs. Specifically, we concatenate the logits from each base model into a single feature vector:

$$\mathbf{z} = [\mathbf{z}_1; \mathbf{z}_2; \ldots; \mathbf{z}_M],$$

and train a logistic regression classifier to predict the final label:

$$\hat{y} = \sigma(\mathbf{w}^\top \mathbf{z} + b).$$

The meta-learner is trained on the training set logits. While stacking allows for learning model-specific weights, its performance can be sensitive to validation set size and overfitting.

Model	**Dev F1**	**Test F1**
BERT Baseline	0.6867	0.7123
BERT + CNN	0.6963	0.7056
BERT + BiLSTM	0.7070	0.7237
Hard Voting	**0.7120**	**0.7258**
Soft Voting	0.7064	0.7225
Max Voting	0.7001	0.7232
Stacking	0.6933	0.7154

Table 4: F1 scores on the dev and test sets for the baseline, architectural extensions, and ensemble methods.

4.3 Error Analysis

We conduct an error analysis on our best-performing system, the hard voting ensemble, to identify systematic failure modes and better understand the trade-offs introduced by our extensions. We manually inspected a representative subset of false positives and false negatives from the test set and grouped them into recurring categories. We additionally compare errors made by the ensemble to those made by the published BERT baseline to highlight cases where extensions improve or degrade performance.

False Positives. The most common false positive category arises from **profanity or insults without a protected target**. In these cases, the model over-relies on strong profanity or aggressive tone as a signal for hate speech, even when the insult is directed at an individual rather than a protected group. This category accounts for approximately 20–25% of false positives. Examples include:

> “Nobody gives a f*** what you do b****!”

Another frequent false positive type involves **reclaimed or in-group slur usage**. Here, slurs appear in self-referential, humorous, or in-group contexts rather than being used to demean a protected class. The model struggles to infer speaker identity and intent, leading to over-prediction of hate speech. This category represents roughly 15–20% of false positives. Representative examples include:

> “this n***** a clone”

Finally, the ensemble incorrectly labels **quoted or reported hate speech** as hateful content. In these cases, offensive language is cited in order to describe, report, or condemn it rather than to endorse it. These meta-context errors account for approximately 10–15% of false positives. Examples include:

> “Disabled people are disgusting is a comment that should not be allowed on any social network”

False Negatives. The most prevalent false negative category consists of **implicit hate without explicit slurs**, where statements express exclusion, inferiority, or hostility toward a protected group without relying on strong profanity or commonly recognized slurs. These cases are challenging due to their subtlety and account for approximately 45–50% of false negatives. Examples include:

> “It is not wrong to say that black people are inferior to whites”

Another notable false negative category involves **slur obfuscation and creative spelling**, where offensive terms are masked using symbols or nonstandard spellings to evade detection. These instances represent roughly 10–15% of false negatives. For example:

> "Our city is getting flooded with n!****"

Extension vs. Baseline Comparison. Our extension corrects several errors made by the published BERT baseline, particularly in cases involving **explicit gendered or racial slurs**. In these instances, the ensemble correctly predicts hate speech while the baseline predicts non-hate, indicating improved sensitivity to overtly offensive lexical cues:

> "@user c**** c**** w*** w***"
> "Go change your tampon m*****!"

In instances involving **Mentions/Placeholders**, our extended model correctly predicts non-hate speech, whereas the baseline misclassifies these examples as hateful. This suggests that our model better captures contextual cues and effectively treats mentions and placeholders—so common in social media—as noise, thereby avoiding false positives.

> "@user the greatest team we ever had are leaving us one after the other"

Conversely, the extensions also introduce new errors not made by the baseline. In particular, the ensemble over-predicts hate speech in cases involving **sexual vulgarity** without a protected target, where the baseline correctly predicts non-hate:

> "@user I'm finna say f*** it and get a crispy a** jogging suit"

Additionally, the ensemble occasionally under-predicts hate speech in cases involving **explicit advocacy of violence or death toward protected groups**, where the baseline correctly identifies hateful content:

> "I applaud the m***** of refugees"

Overall, this analysis reveals a clear trade-off introduced by the extensions. While ensembling increases sensitivity to offensive and slur-based language, it also amplifies reliance on surface-level lexical cues. This results in higher false positive rates for non-targeted vulgarity and meta-context cases. The baseline model, while more conservative, avoids some of these errors at the cost of lower overall performance.

5 Conclusion

In this term project, we investigated the task of binary hate speech detection using the MetaHate dataset, a large-scale and diverse benchmark that unifies multiple hate speech corpora. We implemented a published BERT-based baseline and explored several architectural and ensembling extensions, including BERT+CNN, BERT+BiLSTM, and multiple ensemble strategies such as hard voting, soft voting, max voting, and stacking. All models were evaluated using the F1 score for the hate speech class, which is particularly appropriate given the dataset's strong class imbalance.

Among the single-model architectures, the BERT+BiLSTM extension achieved the strongest performance, demonstrating that explicitly modeling sequential dependencies on top of BERT embeddings improves detection of complex and multi-clause hate speech. Ensembling further improved performance, with the hard voting ensemble achieving the best overall test F1 score of 0.7258. This indicates that combining models with complementary error patterns can yield modest but consistent gains in robustness and generalization.

However, none of our implementations reached state-of-the-art performance as reported in the original MetaHate paper. This discrepancy is primarily due to differences in training scale. While the published model used over 1.1 million examples, our experiments were conducted on a 3% stratified subsample of the dataset due to computational constraints. From a statistical learning perspective, reduced training data limits a model's ability to fully capture rare and nuanced hate speech patterns, leading to higher generalization error. Despite this limitation, our results demonstrate that meaningful performance improvements are achievable even in low-resource settings through careful architectural design and ensembling.

Overall, this project highlights the importance of data scale, model architecture, and error-aware evaluation in hate speech detection, and underscores the trade-offs between sensitivity to offensive language and robustness to contextual nuance.

Acknowledgments

We would like to sincerely thank Prof. Yatskar and TA Anirudh for their constant support throughout this project. We are truly grateful for their valuable guidance, insightful suggestions, and thoughtful feedback.

References

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english.

Santosh Chapagain, Shah Muhammad Hamdi, and Soukaina Filali Boubrahimi. 2025. Advancing hate speech detection with transformers: Insights from the metahate. *Deviant Dynamics in Digital Spaces*.

M. Hossin and M. N. Sulaiman. 2015. A review on evaluation metrics for data classification evaluation. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 5(2):1–11.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2022. Bert-based ensemble approaches for hate speech detection.

Paloma Piot, Patricia Martín-Rodilla, and Javier Parapar. 2024. Metahate: A dataset for unifying efforts on hate speech detection. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):2025–2039.

Gil Ramos, Fernando Batista, Ricardo Ribeiro, Pedro Fialho, Sérgio Moro, António Fonseca, Rita Guerra, Paula Carvalho, Catarina Marques, and Cláudia Silva. 2024. A comprehensive review on automatic hate speech detection in the age of the transformer. *Social Network Analysis and Mining*.

Željko Đ. Vujović. 2021. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):1–8.