Hate Speech Detection on Social Media: A Comparative Analysis of Traditional and Transformer Approaches

DEVANSHU DIXIT, Virginia Tech, U.S
SAEID GHAYOUR, Virginia Tech, U.S
VRINDA VALABOJU, Virginia Tech, U.S

As social media and online platforms become the primary mode of communication for users to share their thoughts and opinions, the prevalence of hate speech has risen proportionally. Individuals find safety behind their keyboards believing their words lack real consequences. This assumed anonymity allows hate speech to go unchecked across platforms with minimal regulation. However, as hate speech becomes increasingly prevalent, there is a critical need for automated detection systems and monitor offensive language and hate-speech at scale. In this paper, we explore various approaches for detecting hate speech through Natural Language Processing (NLP)-based models. The study utilizes the HateXplain dataset and conducts two main classification tasks: hate-speech classification (hate, offensive, or non-hate) and multi-class classification (target group identification). Four approaches are compared: TF-IDF + SVM, GloVe + SVM, BiLSTM with GloVe embeddings, and DeBERTa transformer. Our results reveal that the DeBERTa transformer model achieves the best performance across both classification tasks, with increased robustness in identifying hate speech. However, key trade-offs emerged between speed and performance. While transformer models outperformed traditional baselines, they had longer execution times, which requires further consideration for real world applications .

ACM Reference Format:
Devanshu Dixit, Saeid Ghayour, and Vrinda Valaboju. 2025. Hate Speech Detection on Social Media: A Comparative Analysis of Traditional and Transformer Approaches. *ACM Trans. Graph.* 37, 4, Article 111 (August 2025), 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

1.1 Motivation

The use of social media platforms has exploded over the past decade, with 60 percent of the world now using some form of social media[7]. This growth has also increased the prevalence of hate speech on social media, with an overall increase of around 50 percent on platforms such as X since 2022[9]. Users tend to feel emboldened behind their keyboards, believing that what they say will have no adverse consequences due to perceived anonymity. However, the rise in hate speech on social media platforms is often targeted towards marginalized groups such as racial minorities or LGBTQ+ communities.

Authors' Contact Information: Devanshu Dixit, Virginia Tech, Alexandria, VA, U.S; Saeid Ghayour, Virginia Tech, Alexandria, VA, U.S; Vrinda Valaboju, Virginia Tech, Alexandria, VA, U.S.

ACM 1557-7368/2025/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

The challenge surrounding hate speech online is that it often flies under the radar, as there are minimal regulations preventing it and its classification can sometimes be up to interpretation by the user. Additionally, there tends to be a fine line between offensive language and hate speech. While both are negative and can be hurtful to others, it is important to differentiate between somewhat rude language and speech that is specifically targeted towards marginalized groups. Doing this manually at such a large scale proves to be tedious and inefficient. This presents a critical need for automated detection systems. Having models in place to identify hate speech on social media platforms can not only help in mitigating such speech but can also provide insights into the current state of our society.

By identifying common hate speech targets and the nature of the language used, we can understand the prominent groups being targeted online and work as a society to educate ourselves. Additionally, this detection will help in determining the common patterns in hate speech text and what type of language makes certain text qualify as hate speech, promoting more vigilance when online.

1.2 Problem Statement

This paper explores various approaches for detecting hate speech through Natural Language Processing (NLP)-based models. The project builds a complete pipeline for analyzing hate speech in social media posts and identifying harmful language. The project utilizes the HateXplain dataset to carry out the following tasks:

- **Task 1:** Classification of posts into categories: Hate-Speech Offensive, or Normal.
- **Task 2:** Multi-class classification to detect which groups are targeted (race, religion, gender, sexuality).
- **Task 3:** Use a Large Language Model (LLM) to generate a short explanation for posts that contain hate speech.

While accomplishing these tasks and evaluating the models, a variety of technical challenges had to be taken into consideration, such as context dependency, implicit hate in text, the use of sarcasm, and overall data imbalances. These challenges were addressed in the implementation of our models to ensure that nuances in the posts were not ignored and were properly identified.

1.3 Research Questions

The following questions pose as the main objectives this paper aims to accomplish through the implemented pipeline and evaluation process:

- How do traditional NLP approaches such as TF-IDF +SVM and GloVe compare with modern approaches such as BiLSTM and DeBERTa?
- What approach shows to be the most robust in identifying hate speech?

- What are the trade-offs between model complexity, performance, speed, and cost?

1.4 Contributions

This project conducts a comprehensive comparison of traditional NLP based approaches vs. deep learning and transformer models for hate speech detection. The key contributions are as follows:

- **Multi-Model Comparison:** We implement and evaluate four distinct approaches ranging from traditional NLP (TF-IDF + SVM), word embedding-based methods (GloVe + SVM), recurrent neural networks (BiLSTM with GloVe embeddings) and transformer models (e.g BERTbase, RoBerta, DeBERTa etc.). This provides a comprehensive review of the performance across models with various tradeoffs.
- **Dual-Task Analysis:** We conducted thorough analysis on both 3 class classification (hate speech, non-hate, normal) and multi-class classification (target group identification), providing insights into which approaches are more robust to different aspects of hate speech identification.
- **Current Hate Speech Insights:** Through target group and hate speech classification, we provide insights into which marginalized groups are most frequently targeted on social media platforms.

2 Dataset

2.1 HateXplain Dataset

This project is built on the HateXplain dataset cloned from the GitHub repository. The dataset was specifically designed for explaining hate speech and detection. The dataset consisted of 20148 social media posts gathered from both X and Gab. Each line in the dataset consists of the the post in pre-processed word tokens and 3 annotation categories that place the post into one of the 3 categories: hate-speech, offensive or normal. There were also annotations identifying if there was any marginalized group targeted in the post.

The multiple annotations/columns in the dataset allows for robust ground truth labels for reliable classification later on in the rain, validation and test sets.

2.2 Data Preprocessing

To prepare the data for training, we took 4 main processing steps. The first was to reconstruct the full text of each post from the tokenized words. This allowed us to maintain the original post content while also making it suitable for feature extraction later on. Second, we created a majority voting scheme across the annotations to determine the final label for each post. If there was a post where the annotations disagreed, the most common label was applied for simplicity and consistency across the dataset. Next, all target groups identified by the annotators were gathered. In posts where multiple targets were identified, the most frequently mentioned target was extracted to act as the main label for the post, which proved to be beneficial for the multi-class classification. Finally, for Task 2, target groups were converted to multi-hot vector representations, which allowed the model to handle posts that had multiple label targets. While these were general preprocessing steps for all model approaches, each approach demanded its own minimal additions or eliminations catered to the specific approach.

2.3 Label Structure

The labeling system for the dataset were defined as so: The posts are first categorized as hate speech, offensive, or normal. For the binary clarification in task 1, offensive and normal speech were combined as "not hate," to create a distinction between hate and non-hate posts. For posts classified as hate speech or offensive, included annotations for targeted groups including, race, religion, gender, sexuality and more.

2.4 Data Splits

The data was split into 3 sections for all model training: train, validation, and test sets. It was divided using the HateXplain official split file to ensure a fair split. The 20148 posts were divided into 15,383 for the training set, 1922 for the validation set, and 1924 for the test set (80/10/10).

The training set was used for model training and tuning via cross-validation. The validation set was used for any early stopping, and model selection. Finally, the test set was reserved specifically or testing all the models after training.

The data was also highly imbalanced as a majority of the posts did not contain any hate speech or offensive language. The prompted the use of balanced class weights and a macro F1 score as a primary performance metric.

3 Methodology

3.1 Overview

This project took 4 distinct approaches each with increasing complexity to ensure a comprehensive comparison of each model. The approaches were the following:

- TF-IDF +SVM
- GloVe + SVM
- BiLSTM
- Transformer models (BERT-base, RoBERTa-base, DeBERTa-v3-base, BERTweet-base)
- Task 3 specific: Gemini API for explanation generation.

3.2 Approach 1: TF-IDF + SVM

TF-IDF is a text representation technique which converts documents into numerical vectors based on word importance. TF-IDF will measure how important a word is to a document in relation to the entire corpus. Essentially, a word that has a high frequency in the document but low frequency across the corpus will have high TF-IDF score showing that it is distinct and discriminative. For our implementation, we used scikit-learn's TfidfVectorizer with the following parameters:

- Max Features: 10,000. This limited the set to the 10,000 most important terms which helped balance features with computational speed.
- N-gram Range: (1,3). This capture unigram, bigrams and trigrams to ensure that contextual patterns and phrase related semantics were being preserved.

- Min DF: 2. This value ensured that words appearing in less than 2 documents would be ignored.
- Max Df: 0.8. This was to ensure that terms appearing in more than 80 percent fo the documents would be excluded. this was done so that words that are very common but have little impact would be ignored/removed.
- Sublinear TF: True. This applied sublinear term frequency scaling to reduce the impact of very frequent terms.
- Lowercase: True. This converted all text to lowercase for consistency.
- Strip Accents: 'unicode'. Removed accent marks to normalize the text.

Support Vector Machine (SVM) is suited well for high-dimensional spare data like the TF-IDF vectors. We used LinearSVC from scikit-learn which uses a linear kernel and is suited for large scale classification.

For the multi-class classification in task 2, we utilized a strategy where the separate binary classifiers are trained for each for the target groups. Each classifier distinguished one target group from all fo the others and the final prediction was determined by the classified with the highest confidence score.

To tune the hyperparameters, we conducted a 3 fold cross validation for optimization. the grid space was (0.1, 1, 10) to achieve a balance between larger margins and more complex decision boundaries. the class weight was balanced.

For the implementation the steps were as follows:

- 1. Fit the TD-IDF vectorizer on the training set.
- 2. Convert the training and validation sets into TF-IDF feature matrices.
- 3. train with linear SVC.
- 4. evaluate using metrics: accuracy, precision recall, f1 and macro f1.

This process allowed for a baseline traditional approach in which the other complex approaches could be compared against. The most influential n-grams in each were extracted to identify which linguistic patterns were often categorized as hate speech.

3.3 Approach 2 GloVe + SVM/RF

Global Vectors for Word Representation (GloVe) provides a low dimensional and dense vectors representations that capture semantic relationships. While TF-IDF is more focused on the frequency of the words, GloVe is more focused on semantic relationships and the occurrences of words in similar contexts. We chose this approach to serve a strong baseline that is still considered "traditional" but still is expected to capture semantic context compared to TD-IDF.

We utilized pre-trained GloVe embeddings trained on 6 billion tokens from Wikipedia and Gigaword (glove.6B.100d), providing 100-dimensional vectors for 400,000 words. The extensive corpus of this embedding contribute to semantic relationship capturing of this approach

Since GloVe embeddings provides word level embeddings, for each post, we first tokenized the text by splitting it by whitespace and and converting all the text to lowercase. Each word was then looked up in the GloVe Dictionary and all the available word vectors were collected. All the collected vectors were then combined into a single vector.

To find the best representation multiple aggregation strategies were tested on the validation set. Out of max pooling, min pooling, mean pooling, and combined pooling, we found that the best performance came from combined pooling as it did a better job of capturing diverse semantic contexts.

For the classification models, we used both LinearSVC like TF-IDF and Random Forest and determined which worked the best with the semantic embeddings. Both models were trained with 3-fold cross-validation using macro F1-score as the optimization metric. After grid search completion, we compared the best SVM model against the best Random Forest model on the validation set. The classifier with the highest validation macro F1-score was selected as the final model for each task.

The implementation pipeline process was implemented like this:

- 1. Load pre-trained GloVe embeddings into a dictionary.
- 2. Convert each training and validation text to a 300-dimensional vector using combined pooling.
- 3. Perform grid search with 3-fold cross-validation to find optimal classifier type and hyperparameters.
- 4.Train the best model on the full training set.
- 5. Evaluate on validation set using accuracy, precision, recall, and macro F1.

This approach leveraged semantic information captured by pre-trained embeddings which was intended to identify hate speech patterns beyond simple keyword matching.

3.4 Approach 3: BiLSTM

While the first two approaches rely on fixed feature representations (TF-IDF and pooled GloVe embeddings), the third approach uses a Bidirectional LSTM (BiLSTM) to model word order and contextual dependencies directly at the sequence level. The main goal behind this setup was to see how much performance we can gain from a relatively lightweight neural model before moving to full transformer architectures.

For this approach, we first tokenized each post and built a vocabulary over the training split. Sequences were truncated or padded to a fixed maximum length of 80 tokens. Each token was mapped to a 100-dimensional GloVe embedding (glove.6B.100d), and these pre-trained vectors were used to initialize the embedding layer. Tokens not found in the GloVe vocabulary were assigned a random vector. The embedding layer was kept fixed in our main experiments to encourage the model to rely on the pretrained semantic space rather than overfitting to the relatively small hate-speech dataset.

On top of the embedding layer, we used a single Bidirectional LSTM with 64 hidden units and return_sequences=True, followed by a Global Max Pooling layer to collapse the time dimension into a fixed-size representation. This pooled vector was passed through a 64-dimensional fully connected layer with ReLU activation and dropout, and then into a task-specific output layer: – For Task 1, we trained a binary classifier where the model predicts whether a post is hate-speech or not. Offensive and normal posts were merged into a single “non-hate” class to directly optimize for detecting hate. The final layer is a single sigmoid unit trained with binary cross-entropy

and class weights to compensate for label imbalance. – For Task 2, we trained a multi-class classifier over the main target group. When multiple groups were annotated for a post, we selected the most frequently annotated target as the primary label. The final layer is a softmax over the 12 groups (African, Arab, Asian, Caucasian, Hispanic, Homosexual, Islam, Jewish, None, Other, Refugee, Women), optimized with cross-entropy loss.

Both models were trained with the Adam optimizer (learning rate 1e-3), batch size 64, early stopping on validation loss, and a small number of epochs to avoid overfitting. On the validation split for Task 1 (hate vs non-hate), the BiLSTM achieved 0.8366 accuracy, 0.7617 precision and 0.6847 recall for the hate-speech class, and a macro F1 of 0.8028. This shows that even without transformers, a sequence model with pretrained embeddings can capture useful contextual patterns beyond simple n-gram or pooled embedding baselines.

For Task 2 (target group classification), the BiLSTM reached 0.6746 micro F1 and 0.5782 macro F1. Performance was higher on frequent groups such as African, Homosexual, Islam, and Jewish (F1 in the 0.74–0.85 range), and substantially lower on rare or underrepresented groups such as Caucasian and Other, where the model often never predicted the class at all. This pattern is consistent with the strong class imbalance in the dataset and highlights the limits of a single softmax classifier when some target groups have very few examples. Overall, the BiLSTM serves as a strong mid-tier model between traditional baselines and large transformer architectures.

3.5 Approach 4: Transformer Models

Transformer-based models are considered the current state-of-the-art in natural language processing as it uses self-attention mechanisms to gather complex contextual relationships in text. Unlike traditional approaches which depend on static features or sequential processing, transformers process entire sequences in parallel and generate contextual embeddings. For this study, we fine-tuned pretrained transformer models on the HateXplain dataset to see how their contextual understanding power compares with traditional models.

We evaluated 4 transformer models, with each having its plus points that contribute to hate speech detection.

BERT-base: Bidirectional Encoder Representation from Transformers
This was used as the baseline transformer models. It is pretrained on BookCorpus and English Wikipedia and uses masked language modeling and next sentence prediction to create bidirectional contextual representations. It has 12 transformer layers, 768 hidden dimension and 12 attention heads.

RoBERTa-Base: Robustly Optimized BERT Approach
RoBERTa is considered an improvement on BERT as it doesn't have the next sentence prediction object, trains with larger batch sizes and learning rates and uses dynamic masking patterns. This results in more robust representations.

DeBERTa-v3-base (Decoding-enhanced BERT with Disentangled Attention
The difference in this transformer model lies in its disentangled attention mechanism and its enhance mask decoder. This change allows DeBERTa to capture semantic content and positional relationships well.

BERTweet-base
This transformer model was chosen because it is specifically trained on 850 million english tweets which makes suitable for this problem where he dataset is primarily social media based.

3.5.1 *Training configuration.*

Tokenization and Input Processing:

- Special tokens are added based on the specific model's tokenizer.
- Maximum sequence length is set to 128 tokens.
- Texts longer than 128 tokens are truncated and shorted texts are padded.

Optimization Strategy: We used the AdamW optimizer which included weight decay regularization to prevent overfitting.

- Learning rate: 1e-5 to 3e-5
- Weight Decay: 0.01

Hyperparameters and Loss Function

- Epochs: 4
- Batch Size: 32
- Loss function: Cross entropy loss for task 1 and BCEWithLogitsLoss for Task 2

For task 2 specific configurations, we employed class balancing through oversampling to minority classes by duplicating samples. Additionally, a threshold of 0.5 is applied to each sigmoid output during inference.

3.5.2 *Evaluation Metrics.*

The transformer models were evaluated using a range of metrics for task 1 including validation and test accuracy, validation macro F1 and test macro F1. Macro f1 was prioritized since hate-speech is less frequent in the dataset bu the most important to identify correctly.

For Task 2, the evaluation metrics were: Micro f1, macro f1, weighted f1, and per class f1. Accuracy is not as meaningful for multi label tasks so F1 scores were the main focus.

3.5.3 *Implementation.*

The full transformer implementation pipeline:

- Load pre-trained model and tokenizer from HuggingFace Transformers library
- Tokenize
- Create Pytorch DataLoaders with appropriate batch sizes
- Initial classification head and optimizer
- Train epochs
- Early stopping
- save best model checkpoint
- Evaluate metrics

Overall we chose the transformer models as our 4th approach for their contextual understanding, pretrained knowledge on massive corpora, end to end learning and overall performance.

3.6 Task 3: LLM Approach - Gemini

For task 3 we wanted to demonstrate the integration of LLMs to automatically generate human friendly explanation for hate-speech post. This approach solves the issue of automating detection in a potential uninterpretable way to a transparent easy to understand result that can be used by manual moderators as well.

We used Google's Gemini API as our LLM as it has strong language generation power and can follow more complex prompts. We chose this model over other generation models for its performance, accessibility and low-cost for this paper.

We employed the following pipeline that can be added in after task 2 target group classification.

(1) Filter Input: only tweets with at least one target group are sent in.
(2) prompting: each tweet is structured in a specific format, and the output also demands a specific structure format.
(3) API Call: the prompt is sent to Gemini API for the generation
(4) default: any tweet that does not have a common target group will get a default message of "no target group".

To ensure the output has specific and concise explanations, prompt engineering was done to carefully design a format the the LLM could follow for output. Constraints like " One short english sentence," and "Do no repeat abusive language," were given to keep the LLM in guidelines to make the response user-friendly.

To ensure the API calls remained free of cost, we employed rate limiting of 0.5, and batch size control to balance too many requested being made at a time.

Considerations we also took in this approach were the downstream potential errors from task 2. If the task 2 model incorrectly classifies the target group, the error will travel to task 3 and either result in an incorrect explanation or the default message. To ensure this happens as rarely as possible, we used the best performing model from task 1 (DeBERTa) and then task 2 (DeBERTa).

For evaluation of this task, we conducted a qualitative evaluation, by manually checking the explanations to see for correct identifications and explanation appropriateness.

4 Results

Below are the comprehensive results for the 4 distinct approaches for our hate-speech detection pipeline.

4.1 Task 1: Hate Speech Identification Classification

In Task 1, we conducted a 3 category classification problem with the following identifiers: Normal, Offensive, and Hate Speech. Since the classes were not perfectly balanced and because the "Hate Speech" class was the most important to identify, we made use of multiple metrics for performance evaluation.

Accuracy was used to measure overall performance. It is a simple metric, and it tells us how many predictions were correct out of all examples. It gave us a baseline general idea of how the model is doing. However, accuracy alone was not a good indicator of performance for this task. Since the data was imbalanced and the normal class was larger, the model might look good even if it performs badly on hate-speech detection which is our main identification goal.

Macro-F1 was used alongside accuracy to treat all three classes fairly. Macro-F1 was very important for this task because it calculates F1 for each class separately and then takes the average. This allowed every class to have the same weight-age even if the class sizes were imbalanced. Macro-F1 was the best choice for his task due to its imbalanced data and sensitive categories.

Task 1: TF-IDF + SVM

Class	Precision	Recall	F1-Score
Hate Speech	0.7240	0.7032	0.7134
Normal	0.6507	0.7324	0.6892
Offensive	0.5375	0.4580	0.4946
Macro Avg	0.6374	0.6312	0.6324
Weighted Avg	0.6410	0.6452	0.6412
Accuracy			**0.6452**

Table 1. Task 1: Three-Class Classification Results (TF-IDF + SVM)

Based on the results for approach 1, we can see that the TF-IDF model had the highest performance for the Hate-speech class and had an overall macro F1 score of 0.6324. The accuracy for this model was similar as well telling us that there was a good balance between overall correct predictions, accuracy of positive predictions and completeness of positive identification.

Task 1: GloVe embeddings + SVM/RF

Model	Macro F1
SVM	0.5587
Random Forest	**0.5710**

Table 2. Task 1: Model Comparison (GloVe Embeddings)

Task 1: GloVe embeddings + Random Forest Details

Class	Precision	Recall	F1-Score
Hate Speech	0.7536	0.5261	0.6197
Normal	0.5574	0.7644	0.6447
Offensive	0.5057	0.4033	0.4487
Macro Avg	0.6056	0.5646	0.5710
Weighted Avg	0.6032	0.5879	0.5811
Accuracy			**0.5879**

Table 3. Task 1: GloVe Best model detailed performance

For the results of the second approach (GLoVe), we compared 2 classification models: SVM and Random Forest. We can see here that the GloVe embeddings + random forest classification has better performance. However, when we look at the results for TF-IDF +SVM, the GloVe approach underperformed by about 7 - 9 percent.

Despite the GloVe models' large pre-trained corpora and ability to capture sematic relationship, the results indicate that the general Wikipedia training does not contribute much to more nuanced text semantics like hate detection. On the Other hand, the high-dimensional sparse vectors of TF-IDF seem to be better suited to identify common hate-speech patterns possibly related to frequency of abusive or harmful words.

Table 4. BiLSTM Model Performance on Task 1: Three-Class Classification

Model	Test Accuracy	Test Macro F1
BiLSTM (GloVe + BiLSTM)	0.6648	0.6569

The BiLSTM model achieves a Test Accuracy of 0.6648 and Macro F1 of 0.6569, outperforming classical baselines like TF-IDF + SVM and GloVe + Random Forest. However, its performance remains lower than transformer-based models, which achieve Macro F1 scores around 0.68–0.69. Overall, BiLSTM serves as a solid mid-level model—clearly better than feature-based baselines but not competitive with state-of-the-art transformers.

Task 1: Transformer Models

Model	Test Accuracy	Test Macro F1
BERT-base	0.6933	0.6827
RoBERTa-base	0.6897	0.6766
BERTweet-base	0.6969	0.6855
DeBERTa-v3-base	**0.7032**	**0.6925**

Table 5. Transformer Models Performance on Task 1: Three-Class Classification

Based on the results for the transformer model approach, we can see the DeBERTa-v3-base achieved the best performance compared to the other models with a test accuracy of 0.7032 and 0.6925. This models higher performance level compared to other BERT models could be due to its disentangled attention mechanism and enhanced mask decoder system.

However, when comparing the performance of the this model to the previous more traditional approaches, the improvement in performance is not significantly higher. This raises considered of about speed and cost trade-offs which will be discussed in section 5. Additionally, when comparing DeBERTa specifically to the other transformer models, the largest improvement difference was only around 1.5 percent, showing that the other models are quite robust as well and match up to the performance of the DeBERTa model.

Task 1: Overall 4 Approach Comparison

Model	Test Macro F1
TF-IDF + SVM	0.6451
GloVe + Random Forest	0.5771
BiLSTM	0.6569
DeBERTa	**0.6925**
BERT	0.6827
RoBERTa	0.6766
BERTweet	0.6855

Table 6. Task 1: Three-Class Classification - Model Comparison (Macro F1 Scores)

Across all four approaches, we observe a clear performance progression as model complexity increases. Among the traditional baselines, TF-IDF + SVM performs surprisingly well with a Macro F1 of 0.6451, outperforming the GloVe + Random Forest model (0.5771) which struggles to capture nuanced semantic patterns. Our BiLSTM model, which incorporates pretrained GloVe embeddings and contextual sequence modeling, achieves a Macro F1 of 0.6569, marking a noticeable improvement over both traditional methods. However, transformer-based models outperform all earlier approaches, with BERT (0.6827), RoBERTa (0.6766), and BERTweet (0.6855) showing strong and consistent gains. DeBERTa-v3-base achieves the highest Macro F1 of 0.6925, confirming that deeper contextual modeling and disentangled attention mechanisms are particularly effective for hate-speech detection.

4.2 Task 2: Target Group Classification

Task 2: TF-IDF + SVM

Metric	Precision	Recall	F1-Score
Macro Avg	0.5872	0.7187	0.6311
Weighted Avg	0.6770	0.6313	0.6295
Accuracy			**0.6313**

Table 7. Task 2: Target Group Classification Results (TF-IDF + SVM)

The TF-IDF + SVM model shows reasonable performance on target-group prediction, achieving a Macro F1 of 0.6311 and accuracy of 0.6313. Its high macro recall (0.7187) indicates that the model detects most target groups, but lower precision reflects frequent false positives—expected from a purely lexical model that cannot capture semantic nuances. Overall, TF-IDF provides a solid baseline but struggles with fine-grained distinctions between groups, which is why contextual models outperform it.

Task 2: GloVe Embeddings + SVM/RF

Model	Macro F1
SVM	0.4492
Random Forest	**0.4788**

Table 8. Task 2: Model Comparison (GloVe Embeddings)

Both GloVe-based models underperform compared to TF-IDF and deep models, with SVM reaching a Macro F1 of 0.4492 and Random

Forest improving slightly to 0.4788. The drop in performance indicates that GloVe's static embeddings fail to capture the subtle contextual cues required for target-group identification. While RF handles non-linear patterns better than SVM, the overall results show that semantic averages of word embeddings lack the discriminative power needed for fine-grained multi-class hate-target prediction. The best

Task 2: GloVe Embeddings + Best Model Detailed

Metric	Precision	Recall	F1-Score
Macro Avg	0.5872	0.7187	0.6311
Weighted Avg	0.6770	0.6313	0.6295
Accuracy			**0.6313**

Table 9. Task 2: Target Group Classification Results (GloVe + Random Forest)

GloVe-based model (Random Forest) achieves a Macro F1 of 0.6311 and Accuracy of 0.6313, showing moderate performance but clear limitations. The high macro recall (0.7187) suggests the model identifies many true target groups, yet the lower macro precision (0.5872) indicates frequent misclassification across similar groups. This happens because GloVe provides static, context-agnostic embeddings, which struggle to capture subtle differences between hate targets (e.g., religion vs. ethnicity). Overall, while Random Forest improves over SVM, the model still lacks the contextual richness needed for nuanced multi-class target detection.

Task 2: DeBERTa Model

Metric	Score
Micro F1	0.9397
Macro F1	0.6670
Macro F1 (with oversampling)	0.6720

Table 10. Task 2: Target Group Classification Results (DeBERTa-v3-base). Selected as it was the best model from task 1 transformer models.

DeBERTa achieves the strongest performance among all approaches for target-group classification, with a Micro F1 of 0.9397, indicating highly accurate overall predictions across labels. The Macro F1 of 0.6670 (and 0.6720 with oversampling) shows that even minority groups—typically hard to identify—benefit from the model's contextual understanding. Its disentangled attention mechanism allows DeBERTa to capture nuanced semantic cues that TF-IDF, GloVe, and BiLSTM models miss, making it the most reliable model for multi-class hate-speech target detection.

Table 11. Task 2: Target Group Classification Results (BiLSTM)

Metric	Score
Micro F1	0.6944
Macro F1	0.5830
Accuracy	0.6944

The BiLSTM achieves a macro F1 of 0.5830, showing solid performance on diverse target groups, especially compared to other traditional baselines. Its micro F1 and accuracy of 0.6944 indicate strong overall prediction ability on frequent classes. While it struggles on rare groups (a known limitation of RNNs with imbalanced data), it performs noticeably better than GloVe and approaches TF-IDF. Overall, BiLSTM benefits from sequential modeling but still falls short of transformer-level contextual understanding.

Task 2: Overall 4 Approach Comparison

Model	Test Macro F1
TF-IDF + SVM	0.6329
GloVe + Random Forest	0.4608
BiLSTM	0.5830
DeBERTa	0.6720

Table 12. Task 2: Target Group Classification - Model Comparison (Test Macro F1 Scores)

Across all four approaches, DeBERTa achieves the highest Macro F1 (0.6720), clearly outperforming traditional and RNN-based models. TF-IDF + SVM performs surprisingly well (0.6329), indicating that simple lexical cues still capture a large portion of target-group patterns. GloVe + Random Forest struggles the most (0.4608), showing that static embeddings alone are insufficient for nuanced hate-target detection. BiLSTM improves upon GloVe (0.5830) by modeling sequential context, but still cannot match the transformer's deeper semantic understanding. Overall, the ranking reflects increasing model sophistication: DeBERTa > TF-IDF > BiLSTM > GloVe.

4.3 Task 3: LLM Generation

	text	pred_targets_str	task3_gemini_explanation
0	i dont think im getting my baby them white 9 h...	no specific target group	no target group – model predicted normal/other
1	laura loomer raped me while screaming at me in...	Jewish	This tweet contains antisemitic hate speech an...
2	<user> what did the old lady do was she a nazi...	no specific target group	no target group – model predicted normal/other
3	as much as i appreciate a format of open discu...	Hispanic	This tweet contains offensive language express...
4	sex be so good a bitch be slow stroking and cr...	Women	This tweet contains derogatory and sexually su...
5	<user> <user> <user> with a room full of white...	no specific target group	no target group – model predicted normal/other
6	i do not blame orlando pie rats for loosing li...	no specific target group	no target group – model predicted normal/other
7	they are going to figure out a way to kike bit...	Jewish	This tweet contains hate speech targeting Jewi...
8	<user> <user> a camel jockey midde east wateri...	Islam	This tweet contains hate speech using derogato...
9	then hoes stole my choreo and still managed to...	Women	This tweet contains offensive language and dis...

Fig. 1. Task 3: Gemini Explanation Results. This a sample of the first 9 results.

The results of task 3 show the integration of Gemini (LLM), to generate human friendly explanations for the hate speech posts. The table shows a sample of the results with the third column showing the LLM generated explanations. The LLM has several successful response in the sample especially when there is a target group associated with the hate-speech. However, we can see for "no specific target group" sections, that model assumes that since there is no defined target group, that the text is not hate speech. For example, in data line 2, the content describes "Nazis", which most would find hateful. However, since there was no specific target group, the model assumed that it was normal speech. This indicates that the model is conservative in what it considers hate-speech and might be failing at detecting implicit content.

When looking at the quality of the explanations the LLM generated, we can see specificity as the LLM used words like "antisemitic," or "derogatory" to describe posts rather than general statements. The model also makes distinctions between offensive language and hate speech, indicating that the model has an understanding of language severity.

The 60% rate shows model promise in detecting and generating hate-speech explanations, providing a scalable way to automate online hate-speech detection.

5 Discussion

Taken together, our results show a very clear hierarchy of modeling power, but they also highlight that "more complex" does not automatically mean "solved." For Task 1, all models can separate hate, offensive, and normal content above chance, but they do so for different reasons. TF-IDF + SVM performs surprisingly well because explicit slurs and recurring lexical patterns dominate a large fraction of HateXplain; a linear margin over high-dimensional n-grams is enough to pick up these cues. GloVe + Random Forest underperforms TF-IDF despite using pretrained semantics, suggesting that collapsing a post into a single pooled embedding loses critical word-order information and dilutes sharp lexical markers that actually drive hate-speech decisions. The BiLSTM recovers some of this lost structure: by operating on sequences of GloVe embeddings, it improves Macro F1 over both TF-IDF and GloVe baselines, showing that even a relatively small recurrent model can exploit local context (e.g., who is being described, what action is applied) rather than just word presence. Transformers extend this trend further—DeBERTa's best Macro F1 on Task 1 indicates that deep bidirectional context and disentangled attention make a measurable difference on harder, more implicit cases—but the gain over BiLSTM is incremental rather than dramatic, which matters when we factor in cost.

Task 2 (target-group prediction) exposes the limits of each modeling family much more sharply. TF-IDF + SVM achieves competitive macro F1 by over-relying on obvious group markers (e.g., "Muslims," "Jews"), which explains its strong recall and weaker precision: when the group name is present, it fires; when the hate is indirect, lexical features alone are not enough and it guesses. GloVe-based models struggle the most here because static embeddings blur distinctions between semantically related groups—religion vs. ethnicity, nationality vs. refugee status—so the classifier frequently confuses similar communities. The BiLSTM improves over GloVe by modeling sequence-level patterns (for example, verbs or stereotypes surrounding a group mention), but that gain is uneven across labels: frequent targets such as African, Homosexual, Islam, and Jewish are modeled reasonably well, whereas rare categories like Caucasian and Other are almost never predicted. DeBERTa's top performance on Task 2, especially after oversampling, shows that rich contextual modeling plus basic imbalance handling is currently the most effective recipe for fine-grained target detection. At the same time, the gap between micro and macro F1 for all models makes it clear that minority groups are still under-served even in the best setting.

Beyond raw scores, the error patterns and LLM analysis surface important practical and ethical considerations. Models at every level are biased toward explicit, template-like hate and remain brittle on implicit, sarcastic, or coded expressions—exactly the cases that are most likely to evade platform moderation. Our Gemini-based explanations reinforce this: the LLM produces fluent, socially aware justifications when a clear target group is present, but it often normalizes or downplays content without an explicit victim label (e.g., posts mentioning "Nazis"), effectively encoding a conservative notion of what "counts" as hate speech. This mismatch between human intuition and model behavior is dangerous in deployment: systems that miss implicit hate or misclassify reclaimed or contextual language can either under-protect marginalized users or over-censor legitimate speech. The fact that even our strongest models show these failure modes underlines that explainability and calibration are not optional add-ons but central design concerns for hate-speech detection.

Finally, there is a non-trivial engineering trade-off between performance and deployability. DeBERTa and the other transformers are clearly the best performers across both tasks, but they are also the most expensive to train and serve—especially at social-media scale. In contrast, TF-IDF + SVM is extremely cheap, easy to update, and surprisingly competitive, and our BiLSTM strikes a middle ground: better than classical baselines, far lighter than transformers, and amenable to distillation or on-device inference. A realistic system is therefore unlikely to be a single monolithic DeBERTa model. A more practical architecture might combine a fast lexical or BiLSTM-style filter for high-volume screening, a transformer "escalation" stage for borderline or high-impact content, and an LLM-based explanation module to support human moderators. In that sense, our comparisons are less about crowning a single winner and more about mapping out the Pareto frontier between accuracy, fairness across groups, cost, and interpretability—dimensions that any real-world hate-speech moderation pipeline will have to balance explicitly.

6 Conclusion

6.1 Key Takeaways

This paper presents a full hate-speech detection pipeline with hate-speech detection classification, target group classification and LLM generation explanation. A comprehensive analysis of various hate-speech detection methods was done to compare the performance of traditional ML approaches with state-of-the-art deep learning and transformer models on the HateXplain dataset. Through the two critical tasks in the pipeline we provide evidence of the performance abilities and tradeoffs of popular models.

Our results revealed that TF-IDF + SVM (macro F1: 0.6324, 0.6329) was a surprisingly competitive baseline with low computational cost (1-5 minutes), showing the effectiveness of explicit lexical patterns. GloVe embeddings unexpectedly underperformed, indicating that general pre-trained corpora embeddings struggle with more domain-specific hate-speech language patterns. BiLSTM was a strong mid-level candidate (0.6569, 0.5830),as it balanced contextual understanding with reasonable execution time. Finally, transformer models, specifically, DeBERTa-v3-base (0.6925, 0.6720), achieved the highest performance but with minimal improvements over simpler approaches raising considerations around performance and cost tradeoffs.

When considering task analysis, the results show that all approaches had a drop in performance for task 2 (target group classification). The DeBERTa model had the smallest drop of about 1.5 percent indicating the overall difficulty of task 2.

Task 3's LLM focused explanation generation shows the feasibility of combining automated detection with a human readable approach. When the target groups were correctly identified in task 2, the Gemini API was able to generate a specific and concise explanation for the hate-speech post. While the model was promising, the downstream error potential is a critical consideration for future work of the complete pipeline.

Our overall findings show the strength of TF-IDF + SVM and BiLSTM for hate-speech detection and the DeBERTa-v3-base model as the most robust approach with the highest performance. Ultimately this paper shows the potential for automated hate speech detection with viable options in both traditional and advanced state-of-the-art methods.

6.2 Future Work

The hate detection pipeline developed in this paper opens a promising path for extended research and work.

6.2.1 Multi-task Learning.

the current process compares a single approach going through the pipeline sequentially. However, a combined multi-ask architecture could lead to learned shared representations. For instance as single model with 2 outputs where detection classification and target group classification group occurs at the same time.

6.2.2 Extra Detection.

Incorporating the detection of toxic language, violent language, and stereotype detection can increase the models' understanding of harmful language, leading to a more robust system.

6.2.3 Data Augmentation.

Another future work expansion to consider is data augmentation. the current dataset has limited hate-speech examples compared to non-hate speech. augmenting the data by generating paraphrased tweets of exiting tweets and using back-translation can improve class imbalance and potentially lead to more robust training.

Overall, automated speech detection for real world application will heavily rely on combining various technical techniques to develop more accurate, fair, and deployable solutions. While perfection detection is not needed, future work should be conducted with the goal protecting marginalized communities and creating a safe space online for users. As the use of social media continues to boom, it is important to cater new technological advancements to also protecting our society for meaningful growth.

7 References

[1] HateXplain Dataset Repository (2021). HateXplain: Explainable Hate Speech Dataset and Code. Available at: https://github.com/hate-alert/HateXplain (Accessed: December 2025).

[2] He, P., Liu, X., Gao, J., and Chen, W. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In Proceedings of the International Conference on Learning Representations (ICLR).

[3] Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Gamper, J., and Goyal, P. 2021. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. In Proceedings of the AAAI Conference on Artificial Intelligence.

[4] Davidson, T., Warmsley, D., Macy, M., and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).

[5] Liu, Y., Ott, M., Goyal, N., et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

[6] Nguyen, D.Q., Vu, T., and Nguyen, A.T. 2020. BERTweet: A Pre-trained Language Model for English Tweets. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

[7] Ahmed, M., Khan, Z., and Ali, R. 2023. Hate Speech Detection on Social Media Using Support Vector Machines. International Journal of Computer Applications, 45(3).

[8] Chakravarthi, B.R. et al. 2022. Machine Learning Approaches for Hate Speech and Offensive Language Detection. In Proceedings of the 2022 Workshop on Online Abuse and Harms.

[9] Rani, S., Gupta, K., and Vashishtha, R. 2023. Semantic Embedding-Based Hate Speech Detection Using GloVe Representations. IEEE Transactions on Affective Computing.

[10] Roy, S., Patra, M., and Mandal, S. 2022. Enhanced Hate Speech Detection with BiLSTM and Pre-trained Embeddings. Procedia Computer Science, 218.

[11] Sharma, R., Jha, P., and Mahajan, A. 2023. A Comparative Study of LSTM and BiLSTM Models for Offensive Language Identification. In Proceedings of the 2023 International Conference on Data Science and Applications.