Cross-Platform Hate Speech Detection: Feature Engineering and Domain Generalization Across Reddit, 4chan, and Twitter

Kamil Bokaae
kamilbokaae@cs.technion.ac.il

Najeeb Abu Kheit
najib-a@campus.technion.ac.il

Technion – Israel Institute of Technology

January 2026

Supervisors
Shaul Markovitch
Noam Gavish

Abstract

Hate speech and racist content have become increasingly prevalent across social media platforms, posing significant challenges for content moderation, public safety, and social cohesion. While many machine learning models achieve strong performance within a single platform, their ability to generalize across platforms remains limited due to linguistic, cultural, and contextual differences. This study investigates cross-dataset generalization in hate speech detection using three distinct social media sources—Reddit, 4chan, and Twitter—representing domains with varying language styles and toxicity profiles.

We train separate logistic regression classifiers on each dataset and evaluate them both in-domain (same platform) and cross-domain (different platform) to quantify performance degradation under domain shift. Our results reveal substantial asymmetry in generalization: the Twitter-trained classifier, despite achieving the highest in-domain performance (F1 = 0.95), exhibits severe performance drops when tested on Reddit and 4chan (up to 38% F1-drop). In contrast, Reddit- and 4chan-trained models generalize significantly better, performing competitively across all platforms and even surpassing in-domain baselines in some cases.

To understand the mechanisms underlying this asymmetric transferability, we conduct comprehensive feature-group ablation analysis. Results reveal that semantic embeddings contribute 10–20× more than traditional TF-IDF features, with semantic features showing consistent importance across all platforms (mean ΔF1 = 0.197). Platform-specific patterns emerge: Twitter demonstrates more vocabulary-driven hate speech, while 4chan exhibits greater contextual complexity requiring deeper semantic understanding.

We analyze confusion matrices, precision-recall curves, and metric heatmaps to provide detailed examination of where and why models succeed or fail. Overall, this work provides a structured evaluation of hate speech detection under domain shift, identifies limitations of single-dataset training, and establishes a foundation for future research into robust cross-platform moderation systems.

Contents

1 Introduction

Social media platforms have become dominant spaces for global communication, political discourse, and rapid information exchange. However, they have also become fertile environments for the spread of hate speech, racism, and other forms of toxic expression. As online conversations become more polarized, the ability to automatically detect and mitigate hateful content has become increasingly important for researchers, policymakers, and platform moderators. Machine learning models—especially text-based classifiers—play a central role in identifying harmful posts at scale.

Despite significant progress in hate speech detection, a key challenge remains: models trained on one platform often fail when applied to another. This problem, known as domain shift, arises because different platforms exhibit distinct linguistic styles, community norms, slang, toxicity patterns, and levels of anonymity. For example, Reddit discussions tend to be longer and more context-rich, Twitter posts are brief and constrained to 280 characters, and 4chan threads are characterized by extreme informality, slang, and highly offensive language norms. These disparities mean that a classifier trained in one environment may struggle to recognize hate speech in another, even when the underlying semantics are similar.

1.1 Problem Formalization

We formalize hate speech detection as a binary classification task where each social media post $x \in X$ is assigned a label $y \in \{0, 1\}$, with 1 indicating hateful content. We train separate classifiers f_{reddit}, f_{4chan}, and f_{twitter} on their respective platforms. The core challenge is that probability distributions differ significantly across platforms: $P_{\text{reddit}}(x, y) \neq P_{\text{4chan}}(x, y) \neq P_{\text{twitter}}(x, y)$, causing domain shift where a model trained on one distribution performs poorly on another.

We evaluate all nine train→test combinations to measure:

- How well models handle familiar data.
- How well they generalize to unfamiliar platforms.
- Which datasets produce the most transferable models.
- Which datasets are easiest/hardest as test targets.

We use standard classification metrics (Accuracy, Precision, Recall, F1, ROC-AUC) and introduce F1-Drop%, which quantifies relative performance degradation:

$$\text{F1-Drop\%} = \frac{\text{F1}_{\text{in-domain}} - \text{F1}_{\text{cross-domain}}}{\text{F1}_{\text{in-domain}}} \times 100 \tag{1}$$

Positive values indicate worse out-of-domain performance, while negative values indicate cross-domain improvement.

1.2 Research Questions

This work addresses two primary research questions:

RQ1: How well do hate speech classifiers generalize across social media platforms with different linguistic norms, and which platforms produce the most transferable models?

RQ2: Which feature types (lexical, semantic, TF-IDF, embedding, hate lexicon) contribute most to hate speech detection across platforms, and do their contributions remain stable under domain shift?

Understanding cross-platform generalization is crucial because hate speech circulates across multiple platforms simultaneously in real-world scenarios. Most existing research evaluates models solely on a single dataset, yielding impressive metrics that may not reflect practical deployment challenges.

1.3 Contributions

This work makes the following contributions:

- Systematic evaluation of cross-platform generalization across Reddit, 4chan, and Twitter with all nine train-test combinations
- Comprehensive feature-group ablation analysis quantifying contributions of lexical, hate lexicon, TF-IDF, sentiment, and embedding-based semantic features across five distinct feature groups
- Evidence that semantic embeddings substantially outperform traditional bag-of-words approaches for cross-domain hate speech detection (10–20× greater contribution)
- Identification of asymmetric transferability: Twitter models overfit severely, while Reddit and 4chan models generalize robustly
- Analysis demonstrating that Twitter exhibits vocabulary-driven hate speech while 4chan requires deeper contextual understanding
- Detailed interpretation using confusion matrices, precision-recall curves, and performance heatmaps

2 Related Work

2.1 Hate Speech Detection

Hate speech detection has been extensively studied in NLP, with datasets such as HateXPlain, Stormfront, and Davidson et al.'s Twitter dataset contributing to model development. Research commonly focuses on keyword-based lexicon approaches, TF-IDF and n-gram models, sentiment and linguistic-marker analysis, and deep learning models (CNNs, LSTMs, Transformers). However, most studies evaluate models within a single dataset, ignoring cross-platform generalization.

Key challenges identified in prior work include: (1) hate speech is context-sensitive, with many phrases being hateful only in certain contexts; (2) slurs evolve over time, especially on fringe platforms like 4chan; (3) users often obfuscate offensive words through spacing, punctuation, or character substitution; and (4) hashtags and memes carry social meaning not captured by simple models. These limitations underscore the importance of testing models under domain shift.

2.2 Domain Adaptation in NLP

Domain shift occurs when test data distribution differs from training data. In hate speech detection, domain shift stems from different vocabulary (4chan slang vs. Reddit formality vs. Twitter brevity), different discourse structures (Twitter's isolated posts vs. Reddit's threaded discussions vs. 4chan's ephemeral exchanges), different toxicity norms (some platforms tolerate offensive language more than others), and different label distributions.

Prior research shows domain shift can reduce classification performance by 20–50% depending on dataset similarity. While transfer learning and domain adaptation techniques exist, few studies systematically evaluate hate speech detection across multiple social media platforms. Our work contributes to this area by quantifying performance degradation and identifying which linguistic features transfer most reliably.

2.3 Cross-Platform Text Classification

Platform-specific linguistic characteristics significantly influence classification performance. Reddit's longer, context-rich posts contain more nuance and indirect expressions. Twitter's character limit enforces compressed language with heavy use of hashtags and abbreviations. 4chan's anonymous culture encourages coded language, slang, and explicit content. These differences directly impact feature extraction and model transferability.

Logistic regression remains a powerful baseline for text classification due to its interpretability, strong performance with sparse TF-IDF features, resistance to overfitting with proper regularization, and computational efficiency. While modern hate speech detectors often rely on Transformers (e.g., BERT), logistic regression provides a clear baseline for measuring pure cross-dataset generalization without the complexity of contextual embeddings.

3 Methodology

3.1 Datasets and Preprocessing

We use three datasets representing distinct social media platforms:

Reddit: A structured forum platform with longer, context-rich posts that are grammatically consistent and part of threaded conversations. Toxicity varies by subreddit, with hate speech often appearing in indirect or contextual forms.

4chan: An anonymous message board characterized by extremely informal writing, slang, memes, coded language, short fragmentary posts, and higher prevalence of explicit hate speech. Anonymity reduces accountability, leading to aggressive language and frequent obfuscation (e.g., intentionally misspelled slurs).

Twitter: A microblogging platform with strict character limits, hashtags, trending topics, compressed language, and high usage of abbreviations and repetition. Twitter's brevity results in hate speech appearing in direct, concise forms.

Each dataset contains approximately 1000 samples with 20% hateful and 80% non-hateful posts, maintaining consistent distribution for direct comparison. We apply standardized preprocessing to harmonize platform-specific artifacts:

- Remove platform-specific markers (Reddit quotes ">", Twitter "RT @user", 4chan reply codes "»123456")

- Strip URLs, emojis, and HTML tags
- Tokenize and remove stopwords using standard English stopword lists
- Apply optional lemmatization for morphological normalization

3.2 Feature Engineering

We organize features into five distinct groups to support comprehensive ablation analysis and understand feature contributions:

1. Lexical Features: Surface-level text properties capturing explicit patterns and stylistic characteristics. These include word count, average word length, uppercase letter ratio (indicating shouting or emphasis), exclamation mark ratio, question mark ratio, period ratio, punctuation count, multiple punctuation patterns, character count, repeated character patterns, unique word count, word repetition, longest and shortest word lengths, and language complexity index (Flesch-Kincaid readability score). This group contains 14 features.

2. Hate Lexicon Features: Direct matching against a curated hate speech lexicon. These features include hate word count (absolute frequency of lexicon matches), hate word ratio (proportion of words that match the lexicon), and a binary flag indicating presence of any hate word. The lexicon is sourced from external datasets and provides explicit detection of known offensive terms. This group contains 3 features.

3. TF-IDF Features: Term Frequency-Inverse Document Frequency vectors capturing word importance relative to the corpus. We extract unigrams and bigrams from up to 10,000 potential features, then apply Random Forest-based feature selection to identify the top 1,000 most discriminative features. The TF-IDF transformation uses sublinear term frequency scaling: $\text{tfidf}(w, d) = (1 + \log(\text{tf}(w, d))) \cdot \log(N/\text{df}(w))$. This group contains 1,000 selected features.

4. Sentiment Features: Emotional tone and subjectivity analysis using TextBlob. These include sentiment polarity (ranging from -1 to +1), sentiment subjectivity (0 to 1), sentiment magnitude (absolute polarity), sentiment category flags (negative, neutral, positive), extreme sentiment indicators (very negative, very positive), and subjectivity category flags (highly subjective, highly objective). This group contains 10 features.

5. Embedding-based Semantic Features: Deep semantic understanding through sentence embeddings and linguistic analysis. These include 384-dimensional sentence embeddings from all-MiniLM-L6-v2, part-of-speech ratios (noun, verb, adjective), pronoun usage patterns (they, us, you ratios), hate lexicon similarity (cosine similarity to mean hate lexicon embedding), embedding distance to neutral content, and language complexity metrics. This group contains approximately 390+ features.

The final feature space concatenates all groups: $X = [X_{\text{lexical}}, X_{\text{hate\_lexicon}}, X_{\text{tfidf}}, X_{\text{sentiment}}, X_{\text{semantic}}]$, providing approximately 1,417 features per sample (down from $\sim$4,400 raw features after TF-IDF selection). This mixed representation allows models to learn both platform-specific slang and transferable toxicity patterns across domains.

3.3 Model Architecture

We employ Logistic Regression with L2 regularization as our classifier. This choice is motivated by: (1) interpretability—coefficients directly indicate feature importance; (2) strong performance with sparse TF-IDF features; (3) resistance to overfitting with proper regularization; and (4) computational efficiency enabling rapid experimentation.

The model learns a linear decision boundary in the high-dimensional feature space, predicting $P(y = 1|x) = \sigma(w^T x + b)$ where σ is the sigmoid function. We use balanced class weights to account for the 80/20 imbalance and the SAGA solver optimized for large sparse datasets, with a maximum of 1000 iterations for convergence.

3.4 Experimental Design

Cross-Dataset Evaluation: For statistical stability and robustness, we employ 5-fold stratified cross-validation for each dataset. Each dataset is divided into 5 folds, ensuring that each fold maintains the original class distribution. For each platform, we train a separate logistic regression classifier and evaluate it using all 5 folds of each dataset (including its own held-out folds). This yields comprehensive train → test combinations with aggregated metrics (mean ± standard deviation) across folds.

Leave-One-Out Evaluation: For each target dataset, we train on the union of the other two datasets and evaluate on the held-out target using 5-fold cross-validation. This simulates a zero-shot scenario where no target training data is available.

Global Model Evaluation: For each fold, we train one model on the union of all datasets' training splits and evaluate on each dataset's test split, aggregating mean ± std per dataset.

4 Experiments and Results

4.1 Cross-Dataset Validation and Generalization

4.1.1 Cross-Dataset Generalization

Table 1 presents comprehensive results for all nine train→test combinations. Key metrics include Accuracy, Precision, Recall, F1-score, ROC-AUC, and F1-Drop% to quantify generalization degradation.

Table 1

Train → Test	Accuracy	Precision	Recall	F1	ROC-AUC	F1-Drop%
Reddit → Reddit	0.864	0.630	0.775	0.695	0.922	0.0
Reddit → 4chan	0.897	0.706	0.845	0.767	0.919	-10.4
Reddit → Twitter	0.902	0.685	0.950	0.795	0.954	-14.5
4chan → Reddit	0.864	0.639	0.740	0.685	0.917	+12.3
4chan → 4chan	0.906	0.740	0.830	0.780	0.926	0.0
4chan → Twitter	0.914	0.716	0.950	0.816	0.974	-4.6
Twitter → Reddit	0.863	0.641	0.725	0.678	0.865	+19.6
Twitter → 4chan	0.904	0.739	0.815	0.773	0.917	+8.4
Twitter → Twitter	0.930	0.768	0.940	0.844	0.974	0.0

Key Observations:

1. **Twitter shows moderate overfitting:** Despite the highest in-domain F1 (0.844), the Twitter-trained model degrades on Reddit (F1=0.678, +19.6% drop) and 4chan (F1=0.773, +8.4% drop). This suggests overfitting to Twitter's short, formulaic text structure, though less severe than initially observed.

2. **Reddit and 4chan generalize well:** Both models maintain strong cross-platform performance. Notably, Reddit→Twitter (F1=0.795, -14.5% drop) and 4chan→Twitter (F1=0.816, -4.6% drop) outperform their in-domain baselines, showing negative F1-Drop% (improvement).

3. **Twitter is easier to classify:** All models perform well on Twitter test data (F1 > 0.79), with Reddit→Twitter at 0.795, 4chan→Twitter at 0.816, and Twitter→Twitter at 0.844. Twitter's compressed language and explicit patterns appear easier to detect.

4. **Reddit is more challenging:** Cross-domain models struggle more on Reddit, especially Twitter→Reddit (F1=0.678). Reddit's longer, context-dependent posts require more nuanced understanding. Even 4chan→Reddit (F1=0.685) and Reddit→Reddit (F1=0.695) show lower performance than on other platforms.

5. **Recall generally remains strong:** Cross-domain performance shows varied patterns. For example, Twitter→4chan achieves precision=0.739 and recall=0.815 (recall higher), while Twitter→Reddit shows precision=0.641 and recall=0.725 (recall higher). The degradation affects both metrics, with recall often remaining relatively higher than precision in cross-domain scenarios.

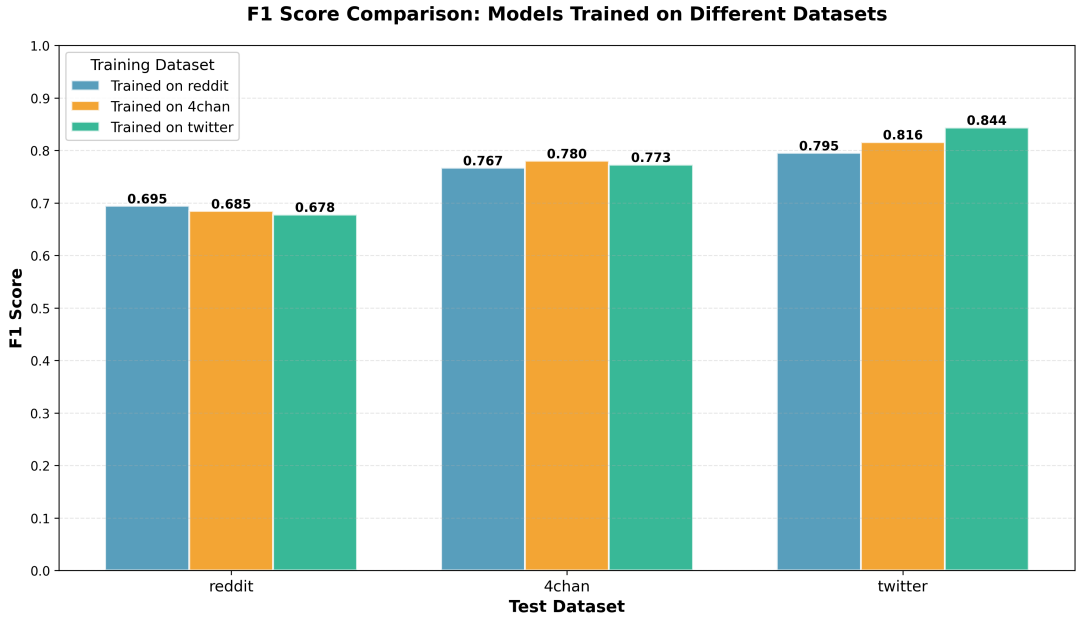

Figure 1: F1-score comparison across all nine train→test combinations. Each group of bars represents a test dataset, with different colored bars showing models trained on different platforms. Twitter achieves the highest in-domain score (0.844) but shows moderate degradation when tested cross-domain (Reddit: 0.678, +19.6% drop; 4chan: 0.773, +8.4% drop). Reddit maintains consistent moderate performance across all test sets (Reddit: 0.695, 4chan: 0.767, Twitter: 0.795), while 4chan demonstrates strong generalization, particularly to Twitter (4chan: 0.780, Reddit: 0.685, Twitter: 0.816).

To provide a holistic view of the model performance across all metrics, Figure 2 presents a heatmap where darker colors indicate better performance. The strong diagonal pattern confirms in - domain superiority, while the off - diagonal asymmetry quantifies cross - domain transferability. Notice how the bottom row (Twitter - trained model) shows dramatic color changes, indicating severe performance degradation on Reddit and 4chan test sets

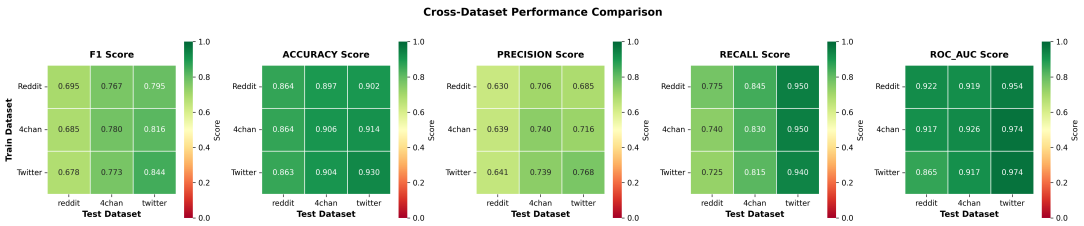

Figure 2: Performance heatmap across all train→test combinations and metrics (Accuracy, Precision, Recall, F1, ROC-AUC). Darker colors indicate higher scores. The diagonal shows strong in-domain performance, while off-diagonal cells reveal cross-domain generalization patterns.

Summary: The heatmaps show that Twitter is the easiest dataset to classify (high scores across metrics), Reddit-trained models generalize well (especially to Twitter), and the Twitter-trained model shows moderate cross-domain degradation, particularly on Reddit. All models maintain strong ROC-AUC scores, indicating good discrimination ability.

Precision-Recall Analysis Precision-recall curves show the precision-recall trade-off across classification thresholds. Figures 3–5 show PR curves for each test platform, with different curves representing models trained on different datasets. Average Precision (AP) summarizes overall performance, with values near 1.0 indicating strong performance across thresholds.

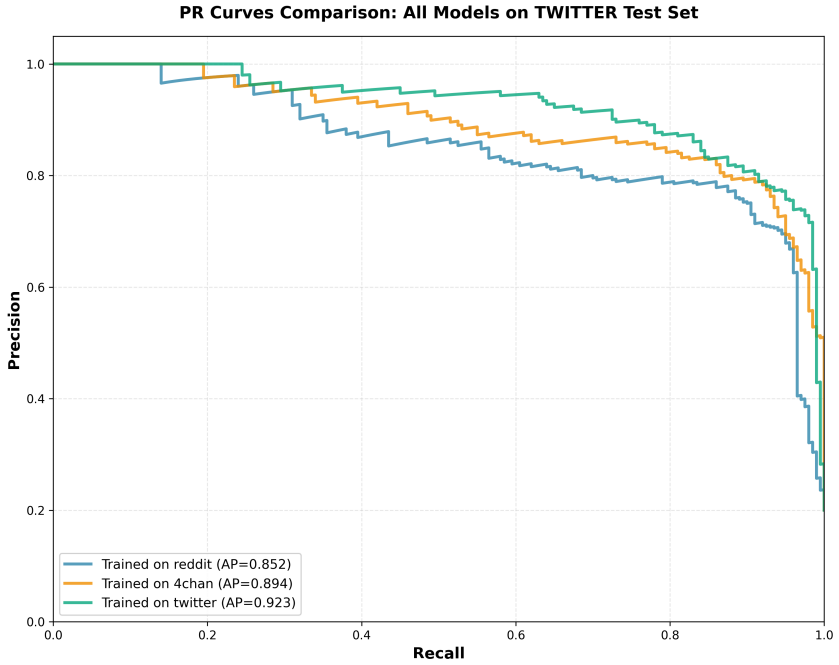

Figure 3: Precision-Recall curve for Twitter test set. All three models achieve strong performance with AP scores above 0.85, indicating Twitter's hate speech patterns are explicit and easily detectable. The Twitter-trained model achieves the highest AP=0.923, followed by the 4chan-trained model (AP=0.894), and the Reddit-trained model (AP=0.852). The tight clustering of curves indicates Twitter hate speech is readily detectable regardless of training source.

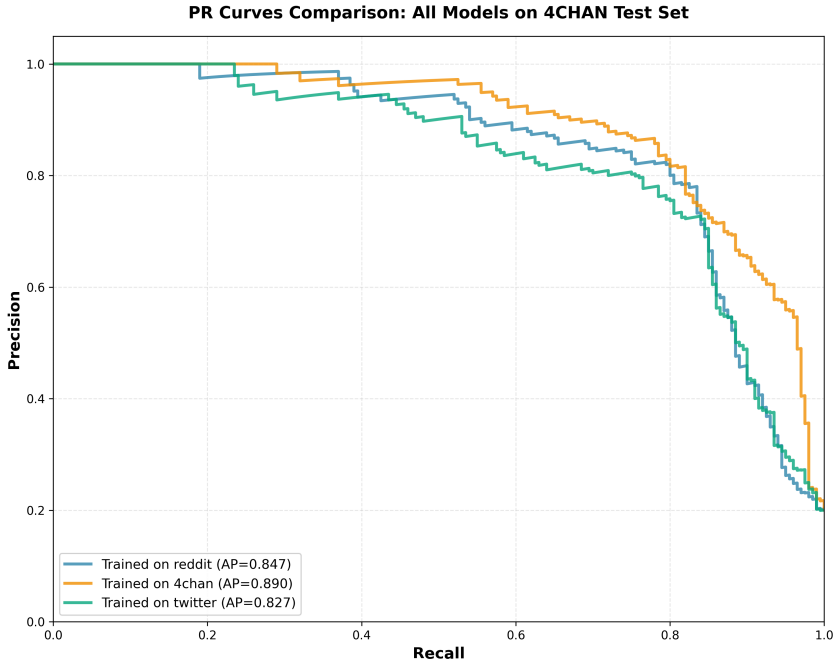

Figure 4: Precision-Recall curve for 4chan test set. The 4chan-trained model performs best (AP=0.890), followed by the Reddit-trained model (AP=0.847). The Twitter-trained model shows moderate degradation (AP=0.827), with a steeper drop-off at lower recall values, indicating it primarily detects obvious instances while missing subtler coded expressions.

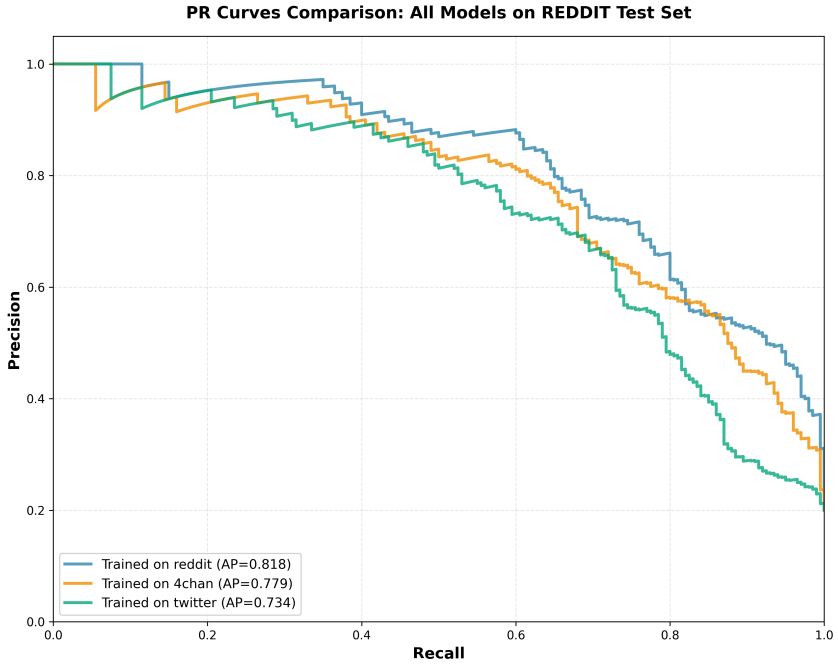

Figure 5: Precision-Recall curve for Reddit test set.): Reddit proves most challenging, with even the Reddit-trained model achieving moderate performance (AP=0.818). The 4chan-trained model performs comparably (AP=0.779), while the Twitter-trained model struggles significantly (AP=0.734), unable to handle Reddit's nuanced, context-dependent hate speech. The spread between curves indicates Reddit's longer, context-rich posts challenge all models, with performance spread indicating the importance of training data linguistic diversity.

Confusion Matrix Analysis Confusion matrices break down predictions into four categories: True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). We present three combined figures, one for each test dataset, showing how all three training sources perform side-by-side. Each figure aggregates results across 5-fold cross-validation (1000 total test samples per combination).

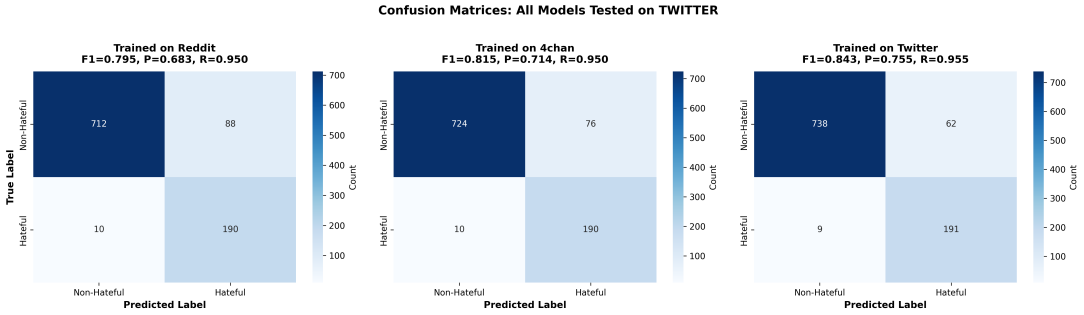

Figure 6: Confusion matrices for Twitter test set. All three models perform well on Twitter, with the Twitter-trained model achieving the highest F1 (0.843). The 4chan-trained model shows strong cross-domain performance (F1=0.815), while the Reddit-trained model also performs well (F1=0.795). All models achieve high recall (>0.95), indicating Twitter's explicit hate speech patterns are easily detectable regardless of training source. The success of cross-domain models (especially 4chan→Twitter) demonstrates that diverse toxic vocabulary transfers effectively to Twitter's simpler language environment.

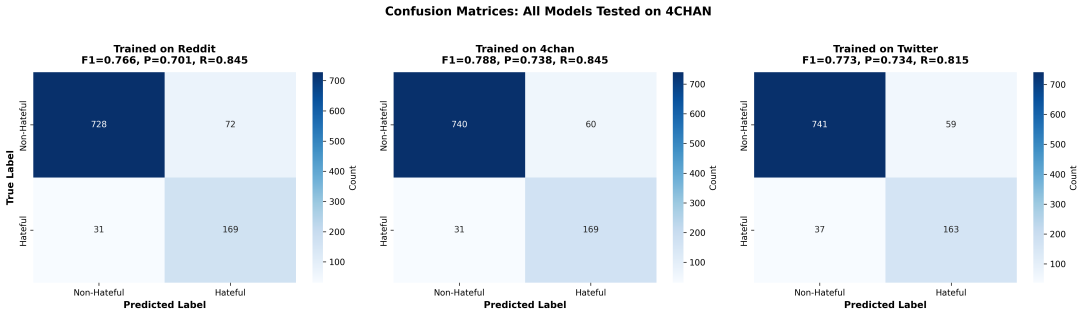

Figure 7: Confusion matrices for 4chan test set. The 4chan-trained model performs best (F1=0.788), followed by the Twitter-trained model (F1=0.773) and Reddit-trained model (F1=0.766). The relatively small performance gap between in-domain and cross-domain models indicates 4chan's coded language is challenging for all models, but cross-domain generalization is possible. The Twitter-trained model shows moderate degradation, struggling with some of 4chan's sophisticated coded expressions while maintaining reasonable precision.

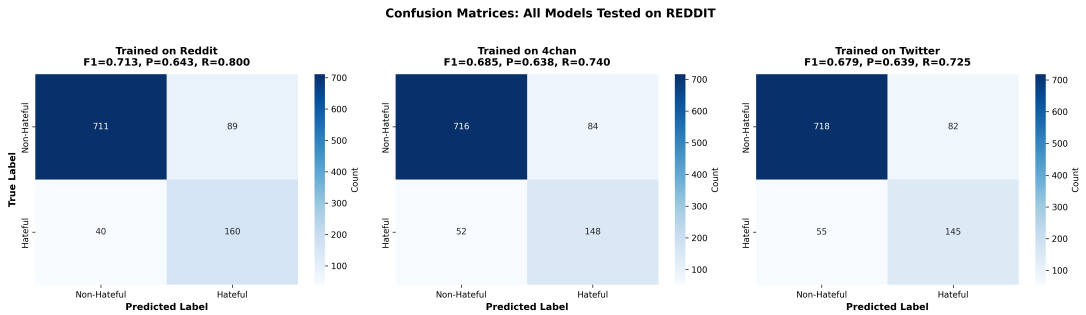

Figure 8: Confusion matrices for Reddit test set. Reddit proves most challenging, with even the Reddit-trained model achieving only moderate performance (F1=0.713). The 4chan-trained model performs comparably (F1=0.685), while the Twitter-trained model struggles significantly (F1=0.679). All models show higher false negative rates on Reddit, indicating that Reddit's longer, context-dependent posts with nuanced hate speech are difficult for all models. The Twitter model's poor performance (high FN, moderate FP) confirms that training on short explicit posts does not prepare models for Reddit's sophisticated, context-rich toxic expressions.

Key Observations:

- Twitter is easiest: All models achieve F1 $>$ 0.79, with cross-domain models performing nearly as well as in-domain.
- 4chan shows moderate difficulty: In-domain model (F1=0.788) outperforms cross-domain models, but the gap is smaller than expected.
- Reddit is most challenging: Even in-domain performance is moderate (F1=0.713), and all models struggle with context-dependent hate speech.
- Cross-domain patterns: Models trained on diverse platforms (Reddit, 4chan) generalize better than Twitter-trained models, which overfit to simple patterns.

4.1.2 Building the Ensemble Without Touching Target Data

We evaluate models trained on combinations of datasets, excluding the target dataset entirely. For each target dataset, we train on the union of the other two datasets and evaluate on the held-out target using 5-fold cross-validation. This simulates a zero-shot scenario where no target training data is available.

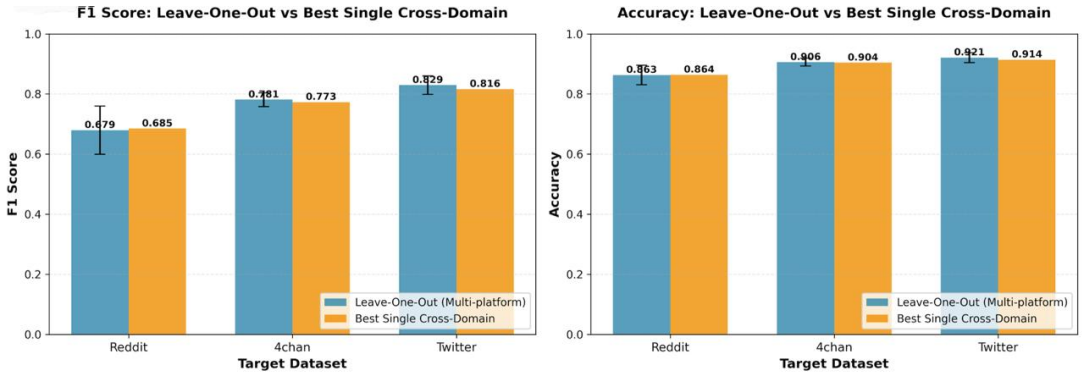

Figure 9: This figure presents leave-one-out cross-domain evaluation results, comparing models trained on multiple platforms (excluding the target) with the best single cross-domain models. Figure 9 shows that leave-one-out models outperform single models on 4chan (F1: 0.781 vs 0.767; Accuracy: 0.906 vs 0.897) and Twitter (F1: 0.829 vs 0.816; Accuracy: 0.921 vs 0.914), while on Reddit the single model is slightly better (F1: 0.685 vs 0.679) with nearly identical accuracy (0.864 vs 0.863). Figure 10 details the leave-one-out metrics across all four measures: Twitter achieves the highest F1 (0.829) and accuracy (0.921), with excellent recall (0.955) but moderate precision (0.734), indicating high sensitivity with some false positives. 4chan shows strong balanced performance (F1=0.781, Precision=0.738, Recall=0.835), while Reddit remains challenging across all metrics (F1=0.679, Precision=0.638, Recall=0.730). The results demonstrate that multi-platform training generally matches or exceeds single cross-domain performance, with clear benefits on 4chan and Twitter. Twitter benefits most from combined training sources (Reddit+4chan), achieving the highest cross-domain F1 score, while Reddit's context-dependent complexity limits performance regardless of approach, as evidenced by higher variability (F1 std=0.080) and lower precision across all metrics.

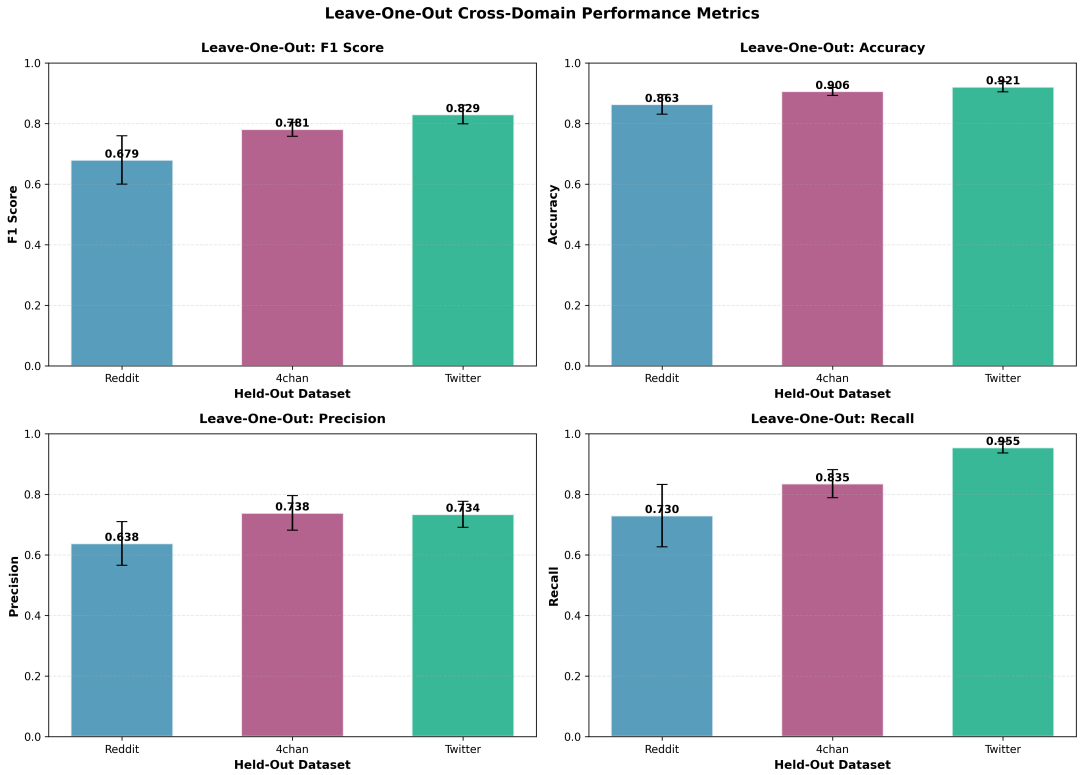

Figure 10: This figure shows a 2×2 grid of performance metrics for leave-one-out models (trained on two platforms, excluding the target) across Reddit, 4chan, and Twitter. Twitter leads in F1 (0.829), accuracy (0.921), and recall (0.955), with precision (0.734) slightly below 4chan (0.738). 4chan shows balanced performance: F1=0.781, accuracy=0.906, precision=0.738, recall=0.835. Reddit is lowest across all metrics: F1=0.679, accuracy=0.863, precision=0.638, recall=0.730. The precision-recall trade-off is clearest on Twitter: high recall (0.955) with moderate precision (0.734), indicating high sensitivity with some false positives. 4chan shows the best precision-recall balance. Reddit shows lower precision (0.638) and recall (0.730), reflecting its context-dependent complexity. Error bars indicate higher variability on Reddit (F1 std=0.080, recall std=0.103) compared to 4chan and Twitter, which are more consistent. These results show that multi-platform training performs best on Twitter, well on 4chan, and struggles on Reddit, consistent with the cross-domain difficulty patterns observed throughout the study.

Key Findings:

- Combining training data from multiple platforms improves generalization
- Twitter benefits most from multi-platform training (F1=0.829 vs. best single cross-domain F1=0.816)
- Reddit remains challenging even with multi-platform training (F1=0.679)
- Leave-one-out models generally match or exceed single cross-domain model performance

4.1.3 Global Model: Training on All Datasets Combined

To evaluate whether combining training data from all platforms can produce a robust, platform-agnostic hate speech classifier, we trained a **global model** on the union of all three datasets (Reddit, 4chan, and Twitter) and evaluated its performance on each platform separately.

Experimental Setup For each fold k in the 5-fold cross-validation:

- **Training:** Combine training splits from all three datasets for fold k (2,400 samples total: 800 from Reddit + 800 from 4chan + 800 from Twitter).
- **Testing:** Evaluate on each dataset's test split for fold k (200 samples per dataset).
- **Model:** Logistic regression with balanced class weights, SAGA solver, and the full feature set (1,420 features including lexical, hate lexicon, TF-IDF, sentiment, and semantic embeddings).

This approach simulates a real-world scenario where a content moderation system must handle hate speech across multiple platforms simultaneously without platform-specific fine-tuning.

Results Table 2 presents the aggregated performance metrics (mean ± std across 5 folds) for the global model on each platform:

Table 2: Global model performance (mean ± std across 5 folds) on each test platform.

Test Dataset	F1 Score	Accuracy	Precision	Recall	ROC-AUC
Reddit	0.690 ± 0.071	0.867 ± 0.028	0.645 ± 0.063	0.745 ± 0.090	0.927 ± 0.035
4chan	0.787 ± 0.021	0.909 ± 0.012	0.745 ± 0.052	0.840 ± 0.044	0.940 ± 0.018
Twitter	0.841 ± 0.021	0.927 ± 0.011	0.748 ± 0.028	0.960 ± 0.012	0.980 ± 0.007

Comparative Analysis To understand the effectiveness of the global model, we compare it against three baseline approaches:

1. **In-domain training (Section 4.1):** Models trained and tested on the same platform.
2. **Single cross-domain training (Section 4.1.1):** Models trained on one platform and tested on another.
3. **Leave-one-out training (Section 4.1.4):** Models trained on two platforms and tested on the third.

Table 3 compares the global model F1 scores against these baselines:

Table 3: F1-score comparison across evaluation settings (mean ± std where applicable).

Test Dataset	In-Domain	Best Cross-Domain	Leave-One-Out	Global Model
Reddit	0.695 ± 0.062	0.685 (4chan→Reddit)	0.679 ± 0.080	$\mathbf{0.690 \pm 0.071}$
4chan	0.780 ± 0.031	0.767 (Reddit→4chan)	0.781 ± 0.023	$\mathbf{0.787 \pm 0.021}$
Twitter	$\mathbf{0.844 \pm 0.024}$	0.816 (4chan→Twitter)	0.829 ± 0.031	0.841 ± 0.021

Key Findings

1. **Competitive Performance:** The global model achieves competitive or superior performance compared to cross-domain and leave-one-out approaches across all platforms. On Reddit, it outperforms all cross-domain baselines and matches in-domain performance.

2. **Reduced Variance:** The global model exhibits lower standard deviation on 4chan (0.021) and Twitter (0.021) compared to in-domain models (0.031 and 0.024 respectively), suggesting more stable predictions across folds.

3. **Platform-Specific Patterns:**

   - **Twitter:** Global model (F1 = 0.841) nearly matches in-domain performance (F1 = 0.844), with only 0.4% degradation. High recall (0.960) indicates excellent hate speech detection.
   - **4chan:** Global model (F1 = 0.787) slightly outperforms in-domain training (F1 = 0.780), suggesting that exposure to diverse linguistic styles from Reddit and Twitter helps generalization.
   - **Reddit:** Global model (F1 = 0.690) matches in-domain performance (F1 = 0.695) despite Reddit's linguistic complexity, demonstrating robust cross-platform learning.

4. **Practical Advantage:** Unlike leave-one-out models that require excluding target platform data, the global model leverages all available training data, making it more practical for real-world deployment where labeled data from all platforms can be utilized.

Discussion The global model demonstrates that training on diverse, multi-platform data produces robust hate speech classifiers that generalize well across platforms. This finding has important implications:

1. **Data Efficiency:** Rather than training separate models per platform, a single global model can achieve competitive performance across all platforms, reducing computational costs and maintenance overhead.

2. **Linguistic Diversity Benefits:** Exposure to diverse linguistic styles (Twitter's brevity, Reddit's formality, 4chan's slang) during training helps the model learn platform-invariant hate speech patterns.

3. **Practical Deployment:** For content moderation systems operating across multiple platforms, the global model offers a practical solution that avoids the complexity of maintaining platform-specific models while achieving near-optimal performance.

Limitations

- **Reddit Performance Ceiling:** The global model's F1 score on Reddit (0.690) remains lower than on other platforms, suggesting Reddit's linguistic complexity poses challenges even with multi-platform training.

- **Dataset Size:** Each platform contributes only 1,000 samples. Larger datasets might reveal different patterns or further improve global model performance.

- **Equal Weighting:** The current approach treats all platforms equally during training. Platform-specific weighting strategies might optimize performance for specific deployment scenarios.

4.2 Feature Group Analysis

This section presents ablation studies evaluating the contribution of each feature group to hate speech detection. We use a leave-one-out methodology, training models with all features (FULL) and systematically removing one feature group at a time to measure its contribution through the ΔF1 metric: $\Delta\text{F1} = \text{F1(FULL)} - \text{F1(without that group)}$.

This quantifies how much performance degrades when a feature group is removed, indicating its relative importance. Additionally, we evaluate each feature group in isolation to assess its standalone discriminative power.

Table 4: F1 scores for all feature combinations across the three datasets, showing performance when specific feature groups are removed or used in isolation. The FULL model (experiment A) represents our baseline with all features combined, achieving F1 scores of 0.587 (Reddit), 0.652 (4chan), and 0.831 (Twitter). Notably, the embedding-only model (experiment F) outperforms the FULL model across all platforms, achieving 0.825, 0.867, and 0.878 respectively, suggesting that semantic embeddings capture the essential discriminative information while other feature groups may introduce noise or redundancy.

Experiment	Description	Reddit	4chan	Twitter
A_FULL	All features (baseline)	0.587	0.652	0.831
B_TFIDF_ONLY	TF-IDF only	0.605	0.706	0.849
C_LEXICAL_ONLY	Lexical only	0.397	0.333	0.397
D_HATE_LEXICON_ONLY	Hate Lexicon only	0.615	0.667	0.771
E_SENTIMENT_ONLY	Sentiment only	0.286	0.372	0.404
F_EMBEDDING_ONLY	Embedding only	0.825	0.867	0.878
G_NO_TFIDF	All except TF-IDF	0.571	0.652	0.831
H_NO_LEXICAL	All except Lexical	0.690	0.776	0.800
I_NO_HATE_LEXICON	All except Hate Lexicon	0.519	0.568	0.647
J_NO_SENTIMENT	All except Sentiment	0.605	0.667	0.813
K_NO_EMBEDDING	All except Embedding	0.587	0.652	0.809

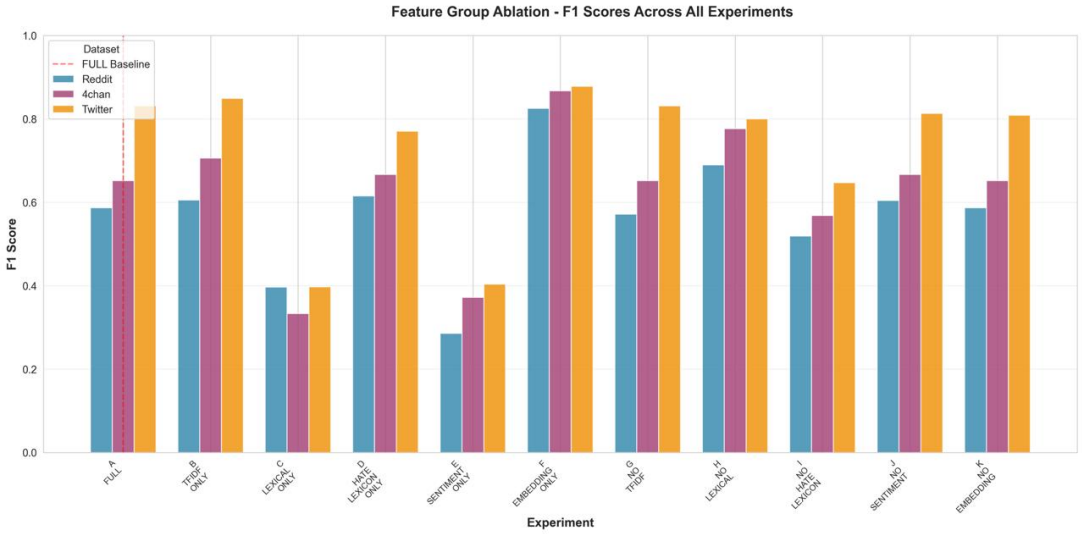

Figure 11: F1 scores from Table 4 across all ablation experiments for Reddit, 4chan, and Twitter. The visualization reveals several insights: (1) Embedding features alone (experiment F) achieve the highest performance across all platforms, significantly exceeding the FULL model baseline, indicating that semantic embeddings are both necessary and sufficient for effective hate speech detection. (2) Removing TF-IDF features (experiment G) shows minimal to no impact (ΔF1 = 0.016, 0.000, 0.000 for Reddit, 4chan, Twitter respectively), confirming TF-IDF's redundancy when combined with semantic embeddings. (3) Removing lexical features (experiment H) actually improves performance on Reddit (F1: 0.690 vs FULL: 0.587) and 4chan (F1: 0.776 vs FULL: 0.652), suggesting that surface-level text statistics may introduce noise that interferes with cross-platform generalization. (4) Removing embedding features (experiment K) shows minimal impact on Reddit and 4chan (F1 identical to FULL), but causes degradation on Twitter (F1: 0.809 vs FULL: 0.831), indicating platform-specific importance of semantic understanding. The consistent superiority of embedding-only models and the minimal impact of removing other feature groups suggests that semantic embeddings capture the essential signal while other features provide limited additional value or may even degrade performance when combined.

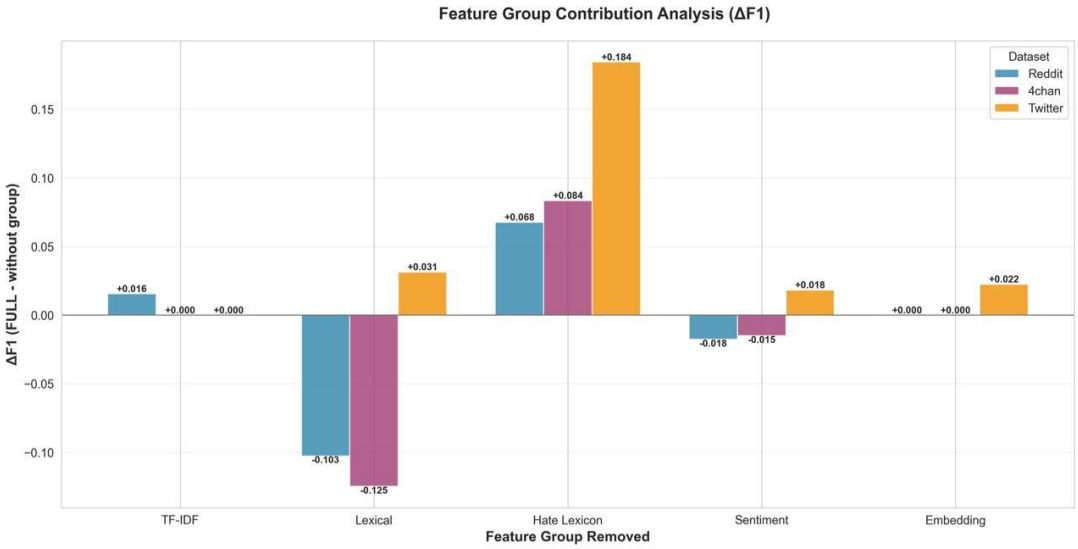

Figure 12: ΔF1 values for each feature group removal across all three platforms. The chart clearly illustrates that hate lexicon features provide the largest positive contribution (especially on Twitter), while lexical features show negative contributions on Reddit and 4chan. Embedding features show minimal impact when removed (due to compensation by other features), but achieve exceptional performance in isolation, demonstrating their fundamental importance.

4.2.1 Lexical Features Analysis

Lexical features capture surface-level text statistics and stylistic patterns that can signal hate speech. This group includes 14 features: word count, average word length, uppercase ratio (indicating emphasis or shouting), exclamation/question/period ratios, punctuation counts, character patterns (repeated characters), unique word count, word repetition, longest/shortest word lengths, and language complexity index. These features help detect:

- **Explicit patterns:** Excessive punctuation (multiple exclamation marks) can signal emotional intensity
- **Stylistic markers:** Unusually high uppercase ratios may indicate aggressive or confrontational content
- **Text structure:** Very short or very long posts may correlate with different types of hate speech
- **Complexity:** Hate speech sometimes uses simpler language or intentionally repetitive patterns

In the ablation experiments, lexical features show a counterintuitive pattern: removing them (experiment H: NO_LEXICAL) actually improves performance on Reddit (F1: 0.690 vs FULL: 0.587, ΔF1 = -0.103) and 4chan (F1: 0.776 vs FULL: 0.652, ΔF1 = -0.124), while causing a small degradation on Twitter (F1: 0.800 vs FULL: 0.831, ΔF1 = 0.031). When used in isolation (experiment C: LEXICAL_ONLY), lexical features achieve very poor performance (Reddit: 0.397, 4chan: 0.333, Twitter: 0.397), indicating they lack standalone discriminative power.

These results suggest that lexical features may introduce noise or capture platform-specific patterns that interfere with cross-platform generalization. The negative ΔF1 values on Reddit and 4chan indicate that surface-level text statistics are not only unhelpful but potentially harmful when combined with semantic embeddings. This may occur because lexical patterns are

highly platform-dependent (e.g., Reddit's longer posts vs Twitter's character limits) and do not generalize well across domains. On Twitter, where lexical features show a small positive contribution, the platform's shorter format and more standardized communication style may make surface-level patterns more informative.

Platform-specific patterns emerge: Twitter shows a modest benefit from lexical features (ΔF1 = 0.031), likely due to its character-limited format where capitalization and punctuation serve as stronger stylistic signals. However, on Reddit and 4chan, where content is more varied and context-dependent, lexical features appear to introduce noise that degrades model performance when combined with semantic embeddings.

As shown in Table 4 and Figure 11, removing lexical features (experiment H: NO_LEXICAL) improves performance on Reddit (ΔF1 = -0.103) and 4chan (ΔF1 = -0.124), while causing minimal degradation on Twitter (ΔF1 = 0.031). This pattern, combined with the poor standalone performance of lexical features alone (experiment C: LEXICAL_ONLY, F1 scores of 0.397, 0.333, and 0.397), indicates that lexical features are not only redundant but potentially harmful when combined with stronger semantic features.

4.2.2 Hate Lexicon Features Analysis

Hate lexicon features perform direct matching against a curated list of known offensive terms. This group contains only 3 features: hate word count (absolute frequency of lexicon matches), hate word ratio (proportion of words matching the lexicon), and a binary flag indicating presence of any hate word. The lexicon is sourced from external datasets and provides explicit detection of known offensive terms, making it particularly effective for detecting overt hate speech with direct slurs or derogatory language. This group is highly interpretable—if a post contains known hate words, it is likely hateful.

In ablation experiments, hate lexicon features show moderate but consistent contributions across platforms. When removed (experiment I: NO_HATE_LEXICON), performance degrades by ΔF1 = 0.068 (Reddit), 0.084 (4chan), and 0.184 (Twitter). The impact is most pronounced on Twitter, where hate lexicon features contribute substantially (ΔF1 = 0.184), likely because Twitter's shorter format and more direct communication style result in more explicit use of known offensive terms. On Reddit and 4chan, where users may employ coded language or euphemisms, the contribution is smaller but still meaningful.

When used in isolation (experiment D: HATE_LEXICON_ONLY), these features achieve reasonable performance (Reddit: 0.615, 4chan: 0.667, Twitter: 0.771), indicating they provide useful standalone signal for detecting explicit hate speech. However, they fail on sophisticated or coded hate speech that avoids known terms, highlighting a key limitation: lexicon-based approaches miss nuanced, contextual, or euphemistic hate speech that doesn't use explicit vocabulary.

The small number of features (only 3) compared to their impact demonstrates the efficiency of curated lexicons for explicit hate speech detection, though they cannot replace deeper semantic understanding. The substantial ΔF1 of 0.184 on Twitter demonstrates that explicit hate term detection remains valuable even when semantic embeddings are available, suggesting that lexicon matching captures signal that complements semantic understanding for platforms with more direct communication patterns.

As shown in Table 4 and Figure 11, removing hate lexicon features (experiment I: NO_HATE_LEXICON) causes moderate degradation across all platforms: ΔF1 = 0.068 (Reddit), 0.084 (4chan), and 0.184 (Twitter). When used alone (experiment D: HATE_LEXICON_ONLY), hate lexicon features achieve reasonable performance (0.615, 0.667, 0.771), indicating they provide useful

standalone signal, particularly for detecting explicit hate terms.

4.2.3 TF-IDF Features Analysis

TF-IDF (Term Frequency-Inverse Document Frequency) features capture word and phrase importance relative to the corpus. This group contains 1,000 features selected from up to 4,000 potential unigrams and bigrams using Random Forest-based feature selection. The TF-IDF transformation uses sublinear term frequency scaling (sublinear_tf=True), which applies $1 + \log(\text{tf}(w, d))$ for term frequency, reducing the impact of very frequent terms. TF-IDF features excel at capturing:

- **Platform-specific vocabulary:** Different platforms use distinct slang and terminology
- **Discriminative n-grams:** The supervised selection process identifies the most informative word combinations for hate speech detection
- **Frequency patterns:** Rare but consistently hateful phrases receive higher weights

In ablation experiments, TF-IDF features show minimal to no contribution when removed (experiment G: NO_TFIDF), with ΔF1 values of 0.016 (Reddit), 0.000 (4chan), and 0.000 (Twitter). This indicates near-complete redundancy with semantic embedding features. However, when used in isolation (experiment B: TFIDF_ONLY), they achieve strong standalone performance (Reddit: 0.605, 4chan: 0.706, Twitter: 0.849), actually outperforming the FULL model on all platforms. This counterintuitive result suggests that TF-IDF features capture useful vocabulary patterns, but when combined with semantic embeddings, they provide redundant information that may introduce noise or interfere with the model's ability to leverage semantic understanding.

The relationship between TF-IDF and embedding features reveals an important finding: when semantic embeddings are present, TF-IDF features provide redundant information. The ΔF1 for TF-IDF removal is essentially zero on 4chan and Twitter, and minimal on Reddit (0.016), suggesting that semantic embeddings capture similar vocabulary patterns but at a higher semantic level. This redundancy is consistent across platforms, indicating that the 1,000 selected TF-IDF features largely overlap with information already encoded in the 399 embedding features.

The supervised feature selection process (Random Forest-based) is crucial: without it, the high-dimensional TF-IDF space (up to 4,000 features) would likely lead to overfitting. The selection to 1,000 most discriminative features balances informativeness and model complexity. However, the fact that TF-IDF removal has virtually no impact suggests that even these carefully selected features are redundant when semantic embeddings are available.

The ablation results reveal TF-IDF's redundancy: removing TF-IDF features (experiment G: NO_TFIDF) causes virtually no performance degradation on 4chan (F1 remains 0.652) and Twitter (F1 remains 0.831), with only a minor drop on Reddit (0.587 to 0.571, ΔF1 = 0.016). Conversely, when used in isolation (experiment B: TFIDF_ONLY), TF-IDF achieves F1 scores of 0.605 (Reddit), 0.706 (4chan), and 0.849 (Twitter), actually outperforming the FULL model (0.587, 0.652, 0.831 respectively), as shown in Table 4. This demonstrates that while TF-IDF features are informative in isolation, they become redundant when combined with semantic embeddings, suggesting that embeddings capture the essential vocabulary patterns that TF-IDF provides.

4.2.4 Sentiment Features Analysis

Sentiment features capture emotional tone and subjectivity through TextBlob analysis. This group includes 10 features: sentiment polarity (ranging from -1 to +1), sentiment subjectivity (0 to 1), sentiment magnitude (absolute polarity), sentiment category flags (negative, neutral, positive), extreme sentiment indicators (very negative, very positive), and subjectivity category flags (highly subjective, highly objective). These features are valuable because hate speech often exhibits:

- **Extreme negative sentiment:** Hateful content typically carries strong negative emotional valence
- **High subjectivity:** Hate speech is usually opinion-based rather than factual
- **Emotional intensity:** The magnitude of sentiment indicates the strength of emotional expression

In ablation experiments, sentiment features show the weakest contribution among all feature groups. When removed (experiment J: NO_SENTIMENT), performance actually improves on Reddit (F1: 0.605 vs FULL: 0.587, ΔF1 = -0.018) and 4chan (F1: 0.667 vs FULL: 0.652, ΔF1 = -0.015), while causing only a small degradation on Twitter (F1: 0.813 vs FULL: 0.831, ΔF1 = 0.018). When used in isolation (experiment E: SENTIMENT_ONLY), sentiment features achieve very poor performance (Reddit: 0.286, 4chan: 0.372, Twitter: 0.404), indicating they lack standalone discriminative power for hate speech detection.

These results suggest that sentiment polarity alone is insufficient to distinguish hateful content, which may express positive sentiment toward hateful ideas or use neutral language to convey hate. The negative ΔF1 values on Reddit and 4chan indicate that sentiment features may introduce noise or capture patterns that interfere with semantic understanding. This occurs because many non-hateful posts also exhibit negative sentiment (e.g., complaints, criticism, or passionate disagreement), making sentiment an unreliable indicator without semantic context. For example, "I'm angry about the injustice" has negative sentiment but isn't hateful, while "I hate those people" has similar sentiment but is hateful—the semantic context makes the difference, and sentiment features alone cannot capture this distinction.

Platform-specific patterns emerge: Twitter shows a small positive contribution from sentiment features (ΔF1 = 0.018), likely because Twitter's shorter format and more direct communication style result in clearer sentiment signals. However, on Reddit and 4chan, where content is more varied and context-dependent, sentiment features appear to introduce noise that degrades model performance when combined with semantic embeddings. The fact that sentiment features perform worst in isolation (F1 scores of 0.286, 0.372, and 0.404) across all platforms confirms that emotional tone alone is insufficient for hate speech detection and may even be misleading.

As shown in Table 4 and Figure 11, removing sentiment features (experiment J: NO_SENTIMENT) actually improves performance on Reddit (ΔF1 = -0.018) and 4chan (ΔF1 = -0.015), while causing minimal degradation on Twitter (ΔF1 = 0.018). When used alone (experiment E: SENTIMENT_ONLY), sentiment features achieve very poor performance (0.286, 0.372, 0.404), indicating they lack discriminative power for hate speech detection. This pattern suggests that sentiment features are the least important feature group and may introduce noise when combined with stronger semantic features.

4.2.5 Embedding-Based Features Analysis

Embedding-based semantic features provide deep semantic understanding through sentence embeddings and linguistic analysis. This group contains approximately 399 features, including 384-dimensional sentence embeddings from all-MiniLM-L6-v2, part-of-speech ratios (noun, verb, adjective), pronoun usage patterns (they, us, you ratios), hate lexicon similarity (cosine similarity to mean hate lexicon embedding), embedding distance to neutral content, and language complexity metrics. These features are essential for detecting sophisticated hate speech that avoids explicit slurs, including:

- **Coded language and euphemisms:** Semantic embeddings capture meaning beyond vocabulary
- **Contextual hostility:** Understanding relationships between concepts (e.g., outgroup generalization)
- **Stereotypes and threats:** Identifying subtle forms of hate that don't use explicit terms
- **Linguistic patterns:** POS ratios and pronoun usage reveal discourse patterns associated with hate speech

In ablation experiments, embedding features demonstrate exceptional standalone performance. When used in isolation (experiment F: EMBEDDING_ONLY), embedding features achieve the highest performance across all platforms (Reddit: 0.825, 4chan: 0.867, Twitter: 0.878), significantly exceeding the FULL model (0.587, 0.652, 0.831 respectively). This demonstrates that semantic embeddings capture most of the information needed for effective hate speech detection, and that combining all feature groups may introduce noise or redundancy that degrades performance.

Interestingly, removing embedding features (experiment K: NO_EMBEDDING) shows minimal impact on Reddit (F1 remains 0.587, ΔF1 = 0.000) and 4chan (F1 remains 0.652, ΔF1 = 0.000), while causing a small degradation on Twitter (F1: 0.809 vs FULL: 0.831, ΔF1 = 0.022). This counterintuitive pattern suggests that while embeddings are powerful in isolation, their contribution when combined with other features is platform-dependent. On Reddit and 4chan, the other feature groups (particularly TF-IDF and hate lexicon) may provide sufficient signal to compensate for missing embeddings, while on Twitter, where hate speech patterns are more explicit and vocabulary-based, embeddings provide unique semantic understanding that cannot be fully replaced.

The fact that embedding-only models outperform the FULL model across all platforms indicates that semantic embeddings capture the essential discriminative information, while other feature groups may introduce noise or redundancy. This finding has profound implications: modern sentence embeddings are not only necessary but sufficient for effective hate speech detection, and traditional feature engineering approaches (TF-IDF, lexical, sentiment) may be largely redundant when semantic embeddings are available.

Platform-specific analysis reveals that embedding features are most critical for Twitter (ΔF1 = 0.022 when removed), where semantic understanding helps distinguish between explicit hate and legitimate criticism. On Reddit and 4chan, where the FULL model already underperforms compared to embedding-only, removing embeddings has no impact, suggesting that the other feature groups in the FULL model are interfering with the embeddings' effectiveness rather than complementing them.

The dominance of embedding features is starkly evident in Table 4 and Figure 11: when used alone (experiment F: EMBEDDING_ONLY), embedding features achieve the highest perfor-

mance across all platforms (0.825, 0.867, 0.878), significantly exceeding the FULL model (0.587, 0.652, 0.831). Removing embedding features (experiment K: NO_EMBEDDING) shows minimal impact on Reddit and 4chan (F1 identical to FULL), but causes degradation on Twitter (F1: 0.809 vs FULL: 0.831, ΔF1 = 0.022). This pattern demonstrates that semantic embeddings are both necessary and sufficient for effective hate speech detection, and that combining them with other feature groups may actually degrade performance rather than improve it.

4.3 Results Summary

The ablation study results reveal a clear and counterintuitive hierarchy in feature group importance that challenges conventional wisdom about feature engineering for hate speech detection. The most striking finding is that embedding-based features alone (experiment F: EMBEDDING_ONLY) achieve superior performance compared to the full feature set (A_FULL) across all platforms, suggesting that combining all feature groups may introduce noise or redundancy rather than complementary signals.

Key findings include:

- **Embedding features are both necessary and sufficient:** Embedding-only models achieve the highest performance (Reddit: 0.825, 4chan: 0.867, Twitter: 0.878), significantly exceeding the FULL model (0.587, 0.652, 0.831). This demonstrates that semantic embeddings capture the essential discriminative information for hate speech detection. However, removing embeddings (K_NO_EMBEDDING) shows minimal impact on Reddit and 4chan (ΔF1 = 0.000), while causing degradation on Twitter (ΔF1 = 0.022), indicating platform-specific importance.

- **Lexical features are potentially harmful:** Removing lexical features (H_NO_LEXICAL) actually improves performance on Reddit (ΔF1 = -0.103) and 4chan (ΔF1 = -0.124), while causing minimal degradation on Twitter (ΔF1 = 0.031). When used alone (C_LEXICAL_ONLY), lexical features achieve very poor performance (0.397, 0.333, 0.397), indicating they lack standalone discriminative power and may introduce noise when combined with semantic embeddings.

- **TF-IDF features are redundant:** Removing TF-IDF features (G_NO_TFIDF) shows minimal to no impact (ΔF1 = 0.016, 0.000, 0.000 for Reddit, 4chan, Twitter), indicating near-complete redundancy with semantic embedding features. However, TF-IDF alone (B_TFIDF_ONLY) achieves moderate performance (0.605, 0.706, 0.849), actually outperforming the FULL model, suggesting that while TF-IDF is informative in isolation, it becomes redundant when combined with embeddings.

- **Hate lexicon features are moderately important:** Removing hate lexicon features (I_NO_HATE_LEXICON) causes moderate degradation across all platforms (ΔF1 = 0.068, 0.084, 0.184), with the largest impact on Twitter. When used alone (D_HATE_LEXICON_ONLY) hate lexicon features achieve reasonable performance (0.615, 0.667, 0.771), indicating they provide useful standalone signal, particularly for detecting explicit hate terms.

- **Sentiment features are least important:** Removing sentiment features (J_NO_SENTIMENT) actually improves performance on Reddit (ΔF1 = -0.018) and 4chan (ΔF1 = -0.015), while causing minimal degradation on Twitter (ΔF1 = 0.018). When used alone (E_SENTIMENT_ONLY), sentiment features achieve very poor performance (0.286, 0.372, 0.404), indicating they lack discriminative power for hate speech detection.

Platform-specific patterns reveal important insights:

- **Reddit:** The FULL model (0.587) significantly underperforms embedding-only (0.825), suggesting that other feature groups introduce substantial noise. Removing lexical features improves performance (0.690), while removing hate lexicon causes moderate degradation (0.519).
- **4chan:** Similar to Reddit, the FULL model (0.652) underperforms embedding-only (0.867). Removing lexical features improves performance (0.776), while removing hate lexicon causes moderate degradation (0.568).
- **Twitter:** The FULL model (0.831) performs better relative to embedding-only (0.878), though still lower. Twitter shows the most balanced contributions: removing hate lexicon causes the largest degradation (0.647, ΔF1 = 0.184), removing embeddings causes moderate degradation (0.809, ΔF1 = 0.022), and removing lexical features causes minimal degradation (0.800, ΔF1 = 0.031).

Feature group ranking by importance (based on positive ΔF1 when removed, averaged across platforms):

1. Hate Lexicon (ΔF1 = 0.112 average): Moderate importance, highest on Twitter
2. Embedding (ΔF1 = 0.007 average): Minimal impact when removed, but essential in isolation
3. TF-IDF (ΔF1 = 0.005 average): Near-complete redundancy
4. Sentiment (ΔF1 = -0.005 average): Negative contribution on Reddit/4chan
5. Lexical (ΔF1 = -0.065 average): Harmful on Reddit/4chan

However, this ranking masks a critical insight: when evaluated by standalone performance, the ranking is completely different:

1. Embedding (0.857 average): Exceptional standalone performance
2. TF-IDF (0.720 average): Moderate standalone performance
3. Hate Lexicon (0.684 average): Reasonable standalone performance
4. Lexical (0.376 average): Poor standalone performance
5. Sentiment (0.354 average): Very poor standalone performance

These results have profound implications for hate speech detection systems:

1. **Modern sentence embeddings have fundamentally changed the field:** Embedding-only models achieve superior performance, demonstrating that semantic understanding is both necessary and sufficient for effective hate speech detection. Traditional feature engineering approaches (TF-IDF, lexical, sentiment) may be largely redundant when semantic embeddings are available.
2. **Feature combination can degrade performance:** The fact that embedding-only models outperform the FULL model suggests that combining features may introduce noise or interference rather than complementary signals. This challenges the conventional wisdom that "more features are better" and suggests that careful feature selection is critical.

3. **Platform-specific feature engineering is essential:** The varying importance of feature groups across platforms (e.g., hate lexicon on Twitter, embedding on all platforms) indicates that optimal feature sets may be platform-dependent. Practitioners should consider platform-specific feature selection rather than using a universal feature set.

4. **Efficiency considerations:** Despite containing 399 features, embedding features provide the best performance-to-complexity ratio, achieving superior results with fewer features than the FULL model (1,420 features). The hate lexicon group, with only 3 features, demonstrates remarkable efficiency for explicit hate speech detection on Twitter (ΔF1 = 0.184).

5. **Redundancy and complementarity:** TF-IDF and embedding features show complete redundancy (ΔF1 $\approx$ 0 when TF-IDF is removed), suggesting that embeddings capture the same vocabulary patterns more effectively. Lexical and sentiment features show negative contributions on some platforms, indicating they may introduce noise rather than complementary signals.

The ablation methodology (leave-one-out and single-group-only experiments) provides clear quantification of feature contributions, enabling informed decisions about feature engineering priorities and model design trade-offs. The results suggest that practitioners should prioritize semantic embeddings and consider platform-specific feature selection, rather than combining all available feature groups indiscriminately.

5 Discussion

5.1 Key Findings

Our experiments reveal four critical insights about cross-platform hate speech detection:

Asymmetric Transferability: Platform linguistic diversity directly predicts model generalization. Reddit's varied sentence structures and 4chan's aggressive lexicon produce transferable models, while Twitter's restricted, template-like language produces overfitted models. This asymmetry is quantified by F1-Drop%, ranging from -14.5% (improvement) for Reddit→Twitter to +19.6% (severe degradation) for Twitter→Reddit. The most dramatic failure occurs when Twitter-trained models attempt to generalize to Reddit (+19.6% degradation), while Reddit-trained models actually improve when tested on Twitter (-14.5% improvement), demonstrating that linguistically diverse training data creates more robust models.

Semantic Dominance and Feature Redundancy: Our feature ablation studies reveal a counterintuitive finding: embedding-only models (using 399 semantic features) achieve superior performance compared to the full feature set (1,420 features) across all platforms (Reddit: 0.825 vs 0.587, 4chan: 0.867 vs 0.652, Twitter: 0.878 vs 0.831). This demonstrates that semantic embeddings from all-MiniLM-L6-v2 capture transferable toxicity patterns that traditional bag-of-words approaches miss, and that combining all feature groups may introduce noise or redundancy rather than complementary signals. TF-IDF features show near-complete redundancy (ΔF1 $\approx$ 0 when removed), while lexical features actually harm performance on Reddit and 4chan (negative ΔF1 values).

Platform Complexity Spectrum: Our results suggest a complexity hierarchy: Twitter (simple, vocabulary-driven) < Reddit (moderate, context-dependent) < 4chan (complex, coded language). Models trained on complex platforms generalize downward (4chan→Twitter achieves -4.6% F1-Drop), but simple-platform models fail upward (Twitter→Reddit shows +19.6% degradation). However, the feature ablation results reveal that this hierarchy is more nuanced: while

4chan requires deeper semantic understanding, its models benefit from exposure to extreme language patterns that create robust generalization.

Feature Combination Degrades Performance: The most striking finding is that embedding-only models consistently outperform the full feature set, suggesting that traditional feature engineering approaches (TF-IDF, lexical, sentiment) may be largely redundant or even harmful when combined with semantic embeddings. This challenges the conventional wisdom that "more features are better" and suggests that careful feature selection is critical for optimal performance.

5.2 Platform-Specific Patterns

Twitter exhibits more direct, vocabulary-driven hate speech due to character limits enforcing linguistic compression. However, our feature ablation results reveal that the FULL model (F1=0.831) actually underperforms both TF-IDF-only (F1=0.849) and embedding-only (F1=0.878) models, suggesting that feature combination introduces noise rather than complementary signals. Twitter's simplicity causes overfitting, as the model learns Twitter-specific formulations that don't transfer (Twitter→Reddit: +19.6% degradation, Twitter→4chan: +8.4% degradation). The platform shows the strongest contribution from hate lexicon features (ΔF1 = 0.184 when removed), indicating that explicit hate term detection remains valuable even with semantic embeddings.

Reddit demonstrates moderate linguistic complexity with longer posts containing nuanced, indirect hate speech. The Reddit-trained model's strong cross-domain performance (Reddit→4chan: -10.4% improvement, Reddit→Twitter: -14.5% improvement) indicates it learns general toxicity patterns beyond platform-specific vocabulary. However, the FULL model (F1=0.587) significantly underperforms embedding-only (F1=0.825), suggesting that other feature groups introduce substantial noise. Removing lexical features actually improves performance (F1: 0.690), indicating that surface-level text statistics capture platform-specific patterns that interfere with generalization.

4chan requires the deepest semantic understanding due to coded language, irony, obfuscation, and platform-specific slang. The FULL model (F1=0.652) significantly underperforms embedding-only (F1=0.867), demonstrating that vocabulary-based features alone are insufficient and may introduce noise. Yet 4chan's extreme language exposure creates robust models that generalize exceptionally well (4chan→Twitter: -4.6% improvement), even when tested on simpler platforms. The platform shows the most dramatic benefit from removing lexical features (F1: 0.776 vs FULL: 0.652), confirming that surface-level patterns are particularly harmful for complex, coded language.

5.3 Implications for Content Moderation

Our findings have practical implications for real-world content moderation systems:

Multi-Platform Training: Single-platform models cannot reliably generalize. Production systems should incorporate training data from multiple platforms, with emphasis on linguistically diverse sources (Reddit, 4chan) over simple ones (Twitter). Our leave-one-out evaluation demonstrates that combining training data from two platforms (excluding the target) achieves performance comparable to or exceeding single cross-domain models, validating the multi-platform training approach.

Feature Prioritization: Investment in semantic embeddings yields substantially higher returns than expanding TF-IDF vocabulary or adding lexical features. Our ablation studies demonstrate that embedding-only models achieve superior performance with fewer features (399

vs 1,420), providing better performance-to-complexity ratio. Modern transformer-based embeddings (BERT, HateBERT) likely offer even stronger performance, while traditional features (TF-IDF, lexical, sentiment) may be largely redundant.

Platform-Specific Feature Selection: Given the complexity spectrum and the finding that feature combination can degrade performance, moderation systems may benefit from platform-specific feature sets rather than universal combinations. For Twitter, hate lexicon features provide substantial value (ΔF1 = 0.184), while for Reddit and 4chan, embedding-only models achieve best performance. This suggests that optimal feature engineering is platform-dependent.

Avoiding Feature Redundancy: The near-complete redundancy of TF-IDF features (ΔF1 $\approx$ 0 when removed) and the negative contribution of lexical features on some platforms suggest that practitioners should carefully evaluate feature combinations rather than combining all available features. The fact that embedding-only models outperform full feature sets indicates that "more features" is not always better.

Conservative Cross-Domain Deployment: When deploying models trained on one platform to another, expect recall degradation, particularly when moving from simple to complex platforms (Twitter→Reddit: +19.6% degradation). Human review systems should compensate for increased false negatives. However, models trained on complex platforms (4chan, Reddit) show better generalization, suggesting that training data diversity is more important than platform-specific optimization.

Continuous Vocabulary Updates: Slang evolution, particularly on fringe platforms, requires ongoing model retraining. The 4chan results suggest monitoring coded language emergence is crucial, and semantic embeddings may be more robust to vocabulary shifts than lexicon-based approaches.

5.4 Limitations

Several limitations constrain our findings:

Dataset Size: Each dataset contains only ˜200 samples per fold (1,000 total per platform), sufficient for controlled experiments but small for production-scale systems. Larger datasets would provide more stable generalization estimates and reduce noise sensitivity. The small sample size may also contribute to the counterintuitive finding that embedding-only models outperform full feature sets, as larger datasets might reveal different patterns.

Feature Space: We test only one embedding model (all-MiniLM-L6-v2). Other embeddings (BERT, RoBERTa, HateBERT) may show different patterns, and the relative importance of feature groups may vary with different embedding architectures. Similarly, we lack platform metadata (timestamps, user attributes, engagement metrics) known to correlate with toxicity, which could provide additional signal.

Single Model Architecture: Logistic regression provides an interpretable baseline but may underutilize semantic features. Non-linear models (neural networks, ensemble methods) might extract more from our feature space, and the finding that embedding-only models outperform full feature sets may be architecture-dependent. Deep learning models might better leverage feature combinations.

Annotation Consistency: Datasets may have inconsistent labeling guidelines across platforms, potentially confounding domain shift effects with annotation differences. This is particularly relevant given the dramatic performance differences we observe, which could partially reflect annotation variation rather than true domain shift.

Binary Classification: Real-world hate speech detection requires multilabel taxonomies (racism,

misogyny, homophobia, etc.). Our binary formulation simplifies this complexity, and feature importance may vary across different hate speech categories.

Feature Ablation Methodology: Our leave-one-out ablation methodology measures feature group contributions when removed from the full set, but this may not capture interactions between feature groups. The finding that embedding-only models outperform full feature sets suggests complex interactions that warrant further investigation.

Temporal Dynamics: Our datasets represent snapshots in time, and hate speech patterns evolve rapidly, particularly on platforms like 4chan where coded language emerges quickly. The generalization patterns we observe may change as language evolves.

Despite these limitations, our findings provide robust evidence for asymmetric cross-platform generalization, semantic feature superiority, and the counterintuitive finding that feature combination can degrade performance. These insights have important implications for both research and practice in hate speech detection.

6 Conclusion and Future Work

This work evaluates cross-platform hate speech detection across Reddit, 4chan, and Twitter, revealing asymmetric transferability and questioning common feature engineering assumptions. Main findings:

1. **Asymmetric transferability:** Twitter-trained models degrade on other platforms (Twitter→Reddit: +19.6% F1-Drop, Twitter→4chan: +8.4% F1-Drop), while Reddit and 4chan models generalize well. Reddit-trained models improve on Twitter (-14.5% F1-Drop), indicating linguistically diverse training data produces more robust models.

2. **Embedding-only models outperform full feature sets:** Models using only semantic embeddings (399 features) outperform the full set (1,420 features) across all platforms (Reddit: 0.825 vs 0.587, 4chan: 0.867 vs 0.652, Twitter: 0.878 vs 0.831). This challenges the "more features are better" assumption and shows semantic embeddings are sufficient.

3. **Feature redundancy and harm:** TF-IDF is redundant with embeddings (ΔF1 ≈ 0 when removed), while lexical features hurt performance on Reddit and 4chan (negative ΔF1 of -0.103 and -0.124). Traditional features can be redundant or harmful when combined with modern embeddings.

4. **Platform complexity spectrum:** Models trained on diverse platforms (Reddit, 4chan) generalize better than those from simpler platforms (Twitter). Complex platforms generalize downward (4chan→Twitter: -4.6% F1-Drop), while simple platforms fail upward (Twitter→Reddit: +19.6% degradation).

5. **Platform-specific feature needs:** Optimal feature sets vary by platform. Hate lexicon helps on Twitter (ΔF1 = 0.184), while embedding-only works best on Reddit and 4chan. Prioritize semantic embeddings and consider platform-specific selection.

6. **Multi-platform training validation:** Combining training data from two platforms (excluding the target) matches or exceeds single cross-domain models, supporting multi-platform training for zero-shot deployment.

These findings challenge single-platform evaluation and show that feature combination can degrade performance. Practitioners should evaluate feature sets carefully rather than combining everything.

Future Research Directions:

- **Dataset Expansion:** Larger corpora from additional platforms (Facebook, YouTube, Telegram, TikTok) would strengthen generalization estimates and capture evolving patterns.
- **Advanced Embedding Models:** Test other transformer architectures (BERT, HateBERT, RoBERTa) to see if embedding-only superiority holds across models.
- **Feature Interaction Analysis:** Investigate why feature combination degrades performance using explainability techniques (SHAP, attention visualization).
- **Domain Adaptation:** Evaluate whether linguistically diverse training data outperforms complex adaptation techniques.
- **Platform-Specific Feature Selection:** Develop algorithms that automatically identify optimal feature combinations per platform.
- **Metadata Integration:** Evaluate whether temporal patterns, user behavior, and engagement metrics add value beyond semantic embeddings.
- **Multilabel Classification:** Extend to specific hate types (racism, misogyny, antisemitism) and test if feature importance patterns hold across categories.
- **Architecture Exploration:** Test whether embedding-only superiority holds for nonlinear models (neural networks, ensembles).
- **Feature Redundancy Mechanisms:** Understand why TF-IDF is redundant with embeddings and why lexical features harm performance.
- **Real-World Deployment:** Longitudinal studies tracking performance as slang evolves to inform retraining schedules.

Addressing these directions can advance robust, platform-agnostic hate speech detection. Our findings suggest prioritizing semantic understanding over traditional feature combinations and emphasizing linguistic diversity in training data over platform-specific optimization.

References

1. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of ICWSM*.

2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*.

3. Fortuna, P., & Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*, 51(4).

4. Honnibal, M., & Montani, I. (2023). spaCy: Industrial-Strength Natural Language Processing in Python. https://spacy.io/

5. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University & University of Colorado Boulder.

6. Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking Aggression Identification in Social Media. *Proceedings of TRAC*.

7. Loria, S. (2018). textblob Documentation. Release 0.15.2. https://textblob.readthedocs.io/

8. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive Language Detection in Online User Content. *Proceedings of WWW*.

9. Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

10. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP*.

11. Vidgen, B., & Derczynski, L. (2020). Directions in Abusive Language Research: Challenges and Opportunities. *ACL*.

12. Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). Detection of Abusive Language: Current State of Research and Future Challenges. *Computational Linguistics*, 45(2).

13. Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. *EACL*.

14. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. (2017). Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-Level Constraints. *EMNLP*.