Architecture-specific preprocessing effects in offensive language detection: Comparing classical, recurrent, and transformer approaches

María Teresa Muñoz Martín
s6461336
m.t.munoz.martin@student.rug.nl

Abstract

This paper explores how text preprocessing interacts with model architecture in offensive language detection. Using the OLID dataset from SemEval-2019 Task 6, I compare three preprocessing levels, *raw*, *clean*, and *aggressive*, across eight models from three main families: classical (SVM, Naive Bayes with TF-IDF), recurrent (BiLSTM with GloVe and FastText), and transformer-based (BERT-base, RoBERTa-base, DeBERTa-v3-base, and the tweet-native BERTweet) under a unified training and evaluation pipeline.

Results reveal distinct family-specific interactions: linear SVM remains largely invariant (macro-F1 0.708–0.711), Naive Bayes degrades under cleaning, BiLSTMs improve substantially with aggressive normalization (from macro-F1 0.724–0.734 raw to 0.754–0.758 aggressive), while transformers exhibit intra-family divergence. BERT-base, RoBERTa-base, and BERTweet-base peak with minimal cleaning (macro-F1 0.803, 0.801, 0.809 respectively), whereas DeBERTa-v3-base uniquely benefits from aggressive preprocessing (macro-F1 0.816 vs. 0.810 clean), consistent with its disentangled attention mechanism tolerating reduced positional variation. Error analysis confirms that BERT/RoBERTa lose contextual cues (capitalization, hashtags) under aggressive normalization, while DeBERTa's gains stem from improved generalization on positionally noisy offensive tweets.

These findings indicate that preprocessing needs reflect architectural inductive biases, including attention mechanisms within transformer families, more than task characteristics alone, with practical implications for choosing preprocessing pipelines based on model architecture in offensive language detection systems.

1 Introduction

Detecting offensive language in social media remains a critical challenge for content moderation systems and user safety. Automated classifiers must handle the informal, noisy nature of social media text while maintaining high accuracy across diverse linguistic expressions. Unlike formal text, tweets contain platform-specific artifacts, like hashtags (`#BlackLivesMatter`), user mentions (`@USER`), URLs, creative spelling, and capitalization patterns, that may encode semantic or pragmatic information relevant to offense detection. The central question is whether these surface features should be preserved as informative signals or removed as noise during preprocessing.

Most work in offensive language detection has pursued improvements in either model architectures or preprocessing techniques, but rarely examines their interaction. Architecture-focused studies (e.g., (Wei et al., 2021) with bi-directional LSTMs and BERT, (Mnassri et al., 2023) with transformer encoders) tend to apply standard preprocessing without exploring its impact in depth, while preprocessing-focused work (e.g., (Akhtar et al., 2015) on text normalization for Twitter) typically evaluates strategies using simpler baseline models. However, few studies systematically investigate how these factors interact (Zampieri et al., 2019b). This gap is significant because different model families encode distinct inductive biases: classical bag-of-words models rely on explicit distributional statistics, recurrent models with frozen embeddings depend on vocabulary coverage for representation quality, and modern transformer architectures leverage subword tokenization and contextualized attention that may exploit or be harmed by aggressive normalization. Without systematic comparison across architectures under controlled preprocessing conditions, it remains unclear whether preprocessing strategies should be task-specific or architecture-specific.

This study addresses the research question: *How do preprocessing strategies interact with model architectures that encode different inductive biases for offensive language detection?* I hypothesize that preprocessing sensitivity will vary systematically by architectural family: classical sparse models may show robustness due to high-dimensional distributional abstraction; recurrent models with frozen embeddings may depend strongly on normalization to maximize pretrained vocabulary coverage; and transformer encoders may exhibit variable sensitivity depending on their attention mechanisms and tokenization schemes.

To isolate preprocessing effects from confounds, I conduct a controlled experiment applying three preprocessing levels: *raw* (no modification), *clean* (removing URLs and user mentions), and *aggressive* (full normalization including lowercasing, hashtag stripping, and punctuation removal), across eight models spanning three architectural families: classical TF-IDF baselines (linear SVM, Multinomial Naive Bayes), bidirectional LSTMs with frozen pretrained embeddings (GloVe-Twitter and FastText), and four transformer encoders fine-tuned under identical optimization settings (BERT-base-uncased, RoBERTa-base, DeBERTa-v3-base, and the tweet-native BERTweet-base). All models are evaluated on the OLID dataset from SemEval-2019 Task 6 (Zampieri et al., 2019b), using the official train/dev/test splits to ensure comparability with shared task results.

The experiments reveal that preprocessing impact varies not only across architectural families but also within them, challenging the assumption that preprocessing needs are task-determined rather than model-determined. Classical models exhibit contrasting behaviors: margin-based SVM remains nearly invariant while probabilistic Naive Bayes degrades under cleaning, reflecting differences in how feature independence assumptions interact with surface variation. Recurrent models show strong preprocessing dependence driven by embedding vocabulary coverage. Most surprisingly, transformer models diverge within their own family: standard-attention encoders prefer minimal cleaning while disentangled-attention architectures benefit from aggressive normalization, indicating that even within state-of-the-art contextual models, attention mechanisms modulate preprocessing sensitivity.

My contributions include: (1) the first systematic comparison isolating preprocessing effects across classical, recurrent, and transformer families under controlled experimental conditions, spanning 24 model-preprocessing configurations evaluated on a common benchmark; (2) mechanistic analysis linking preprocessing sensitivity to architectural properties, vocabulary coverage for recurrent models, conditional independence for probabilistic classifiers, and attention disentanglement for transformers; and (3) practical guidelines demonstrating that preprocessing strategies must be tailored to model architecture rather than assumed universal, with implications for how NLP pipelines should be designed when model families are compared or deployed.

The remainder of this paper is organized as follows. Section 2 reviews related work on offensive language detection, preprocessing strategies, and architectural comparisons. Section 3 describes the OLID dataset and the three preprocessing strategies. Section 4 details the experimental methodology covering classical, recurrent, and transformer models with unified evaluation protocols. Section 5 presents comprehensive results across all configurations, including aggregate metrics and systematic error analysis comparing misclassification patterns across preprocessing strategies for each model family. Section 6 analyzes key findings, connects preprocessing sensitivity to architectural inductive biases through mechanistic interpretation of error patterns, and discusses limitations. Section 7 concludes with implications for preprocessing design in offensive language detection systems and broader NLP applications.

2 Related Work

Offensive language detection on social media has been catalyzed by the OffensEval shared task at SemEval-2019, which introduced the OLID dataset (13,240 annotated tweets) for binary offensive/not-offensive classification (Zampieri et al., 2019b). The winning NULI system fine-tuned

BERT-base and achieved a macro-F1 of 0.829, establishing transformer supremacy (Liu et al., 2019a). However, competitive non-transformer approaches, including BiLSTMs with FastText embeddings (macro-F1 0.799) and CNNs with GloVe-Twitter (0.789), demonstrated that recurrent architectures with pretrained embeddings remained viable alternatives (Badjatiya et al., 2019; Mitrovi'c et al., 2019), as shown in Table 1.

A persistent debate concerns whether Twitter-specific features (hashtags, mentions, URLs, capitalization) should be preserved or removed. Eisenstein argued that non-standard orthography encodes sociolinguistic signals and cautioned against indiscriminate normalization (Eisenstein, 2013), while comparative studies in sentiment analysis reported benefits from aggressive cleaning to reduce vocabulary sparsity (Symeonidis et al., 2018). For offensive language detection specifically, prior work shows mixed results: classical pipelines benefit from URL/mention removal (Davidson et al., 2017), yet hashtags can provide crucial discourse context (Zhang et al., 2018). These conflicting findings motivate controlled comparison of preprocessing strategies across architectures.

Classical TF-IDF baselines remain competitive under appropriate feature engineering. Linear SVMs learn maximum-margin hyperplanes in sparse high-dimensional spaces (Joachims, 2002; Wang and Manning, 2012), while Multinomial Naive Bayes provides a probabilistic approach under conditional independence assumptions (McCallum and Nigam, 1998). Both served as competitive OffensEval baselines despite being outperformed by transformers. Recurrent models with frozen pretrained embeddings offer a middle ground: BiLSTMs with GloVe or FastText capture sequential dependencies while leveraging domain-tuned semantic knowledge (Pennington et al., 2014; Bojanowski et al., 2017). Their effectiveness depends critically on vocabulary coverage, aggressive preprocessing that reduces out-of-vocabulary tokens can substantially improve performance by enabling better exploitation of pretrained representations. However, the trade-off between vocabulary normalization and preservation of informative surface features remains underexplored.

While BERT (Devlin et al., 2019a) and RoBERTa (Liu et al., 2019b) established transformer dominance for offensive language detection, recent architectural innovations introduce distinct inductive biases relevant to preprocessing sensitivity. DeBERTa (Decoding-enhanced BERT with Disentangled Attention) fundamentally modifies the standard transformer attention mechanism by representing tokens with two separate vectors encoding content and relative position independently, rather than summing them as in BERT/RoBERTa (He et al., 2021). Attention weights are computed via disentangled matrices over content-to-content, content-to-position, and position-to-content relationships. This architectural choice may affect robustness to surface variation: if preprocessing collapses positional noise (e.g., “#BlackLivesMatter” → “blacklivesmatter”), DeBERTa’s disentangled representations may benefit by reducing spurious positional cues while preserving semantic content. Conversely, BERTweet is a RoBERTa-base model pretrained exclusively on 850 million English tweets, preserving Twitter conventions (mentions, URLs, emojis) as special tokens (Nguyen et al., 2020). Unlike general-domain BERT trained on Wikipedia and BookCorpus, BERTweet’s pretraining distribution matches social media linguistic characteristics. Whether domain-matched pretraining modulates preprocessing needs remains unclear: a model exposed to noisy tweet text during pretraining may tolerate or exploit surface variation, potentially reducing the necessity for aggressive normalization.

Prior work either optimizes a single architecture with varied preprocessing (Davidson et al., 2017) or compares architectures under fixed preprocessing (Zampieri et al., 2019b), but does not systematically isolate preprocessing effects across model families. This study addresses whether preprocessing recommendations are architecture-dependent by evaluating eight models spanning three families (classical TF-IDF, BiLSTMs with frozen embeddings, four transformer variants with standard/disentangled attention and general/domain-specific pretraining) under three controlled preprocessing strategies on the OLID benchmark.

3 Data

I use the Offensive Language Identification Dataset (OLID) introduced by Zampieri et al. (Zampieri et al., 2019a) for SemEval-2019 Task

System	Architecture	Macro-F1
NULI	BERT-base fine-tuned	0.829
NLPR@SRPOL	BERT + ensemble	0.824
MIDAS	BiLSTM + FastText	0.799
NLPUP	CNN + GloVe-Twitter	0.789

Table 1: Top systems from OffensEval-2019 Subtask A on the OLID dataset, establishing benchmark performance for offensive language detection.

6. The corpus contains English tweets annotated for offensive language at three hierarchical levels. For this study, I focus exclusively on Sub-task A: binary classification of tweets as offensive (OFF) or not offensive (NOT).

The dataset comprises 13,240 tweets split into training (12,240 tweets), development (1,000 tweets), and test (860 tweets) sets. Table 2 presents the distribution across splits. The dataset exhibits moderate class imbalance, with approximately 33% of tweets labeled offensive across all splits. This imbalance reflects realistic social media content distributions but necessitates careful metric selection to avoid inflating performance through majority-class bias.

Split	Total	OFF	NOT	OFF %
Train	12,240	4,048	8,192	33.07
Dev	1,000	352	648	35.20
Test	860	240	620	27.91

Table 2: Dataset statistics for OLID Sub-task A across train, development, and test splits.

Tweets average 125–146 characters in length and contain typical social media characteristics: user mentions (@USER, anonymized for privacy), URLs, hashtags, emojis, and non-standard orthography. Example offensive tweets include explicit insults and threats, while non-offensive tweets range from neutral news commentary to sarcastic political opinions. The boundary between offensive and non-offensive content is context-dependent, particularly for tweets containing profanity in non-hostile contexts or sarcastic criticism of public figures.

I implement three preprocessing strategies uniformly across train, development, and test sets for each strategy to investigate the impact of Twitter-specific feature removal:

Raw: No preprocessing. Text is used exactly as provided in the corpus, preserving all mentions, URLs, hashtags, emojis, and original capitalization. This condition tests whether models can learn from authentic social media language without human intervention.

Clean: I remove URLs (http, www patterns) and user mentions (@USER) while preserving other features. This represents minimal cleaning that removes obvious noise (links provide no textual information; mentions are anonymized) while retaining potentially informative hashtags and emojis. Excessive whitespace is collapsed to single spaces.

Aggressive: I apply extensive normalization: remove URLs and mentions; strip hashtag symbols but keep the text (e.g., #BlackLivesMatter → BlackLivesMatter); remove numbers and special characters; lowercase all text; and collapse whitespace. This condition tests whether maximal simplification improves model performance by reducing feature space dimensionality.

4 Method

I adopt a systematic experimental design to isolate preprocessing effects from model architecture. Each preprocessing strategy (raw, clean, aggressive) is paired with eight models from three families (2 classical baselines, 2 recurrent models with distinct embeddings, 4 transformer variants), yielding 24 configurations. Development set performance guides hyperparameter selection and early stopping, while final evaluation uses the test set.

4.1 Classical baselines: TF-IDF + Linear models

Feature representation: Text is converted to numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization (Salton and Buckley, 1988), extracting both unigrams and bigrams (word-level 1-2 n-grams) to capture individual terms and local context. The vocabulary is limited to 10,000 most frequent n-grams, with terms appearing in fewer than 2 documents discarded. Vectorization parameters are fit on training data only.

Models: I employ (1) Multinomial Naive Bayes, which models text classification probabilistically assuming conditional independence of features (McCallum and Nigam, 1998), and (2) linear SVM with L2 regularization ($C = 1.0$), which finds the maximum-margin hyperplane in TF-IDF space (Joachims, 2002). These baselines test whether sparse feature representations benefit from aggressive feature space reduction. Both models are implemented using scikit-learn (Pedregosa and others, 2011).

4.2 BiLSTM with frozen pre-trained embeddings

To evaluate sequence models dependent on pre-trained vocabulary coverage, I train bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) with frozen word embeddings.

Architecture: Input tweets are tokenized via whitespace splitting and mapped to pre-trained vectors (200-dimensional for GloVe, 300-dimensional for FastText), then padded/truncated to length 64. A BiLSTM layer (hidden size 128) processes sequences bidirectionally, concatenating final forward and backward hidden states into a 256-dimensional representation. Dropout (rate=0.3) precedes a fully connected classification layer (output dim=2) with softmax. Total parameters: $\sim$2.5M (GloVe) and $\sim$3.7M (FastText), with $\sim$340k trainable parameters (embeddings remain frozen to isolate preprocessing effects on vocabulary coverage).The architecture is implemented in PyTorch (Paszke and others, 2019).

Embeddings: I compare two pretrained embeddings with distinct training domains. **GloVe-Twitter** (Pennington et al., 2014) provides 200-dimensional vectors trained on 2 billion tweets (1.2M vocabulary). This domain-specific training captures Twitter conventions such as hashtags, slang, and abbreviations effectively, but suffers from out-of-vocabulary (OOV) issues when applied to unnormalized text. In contrast, **FastText** (Bojanowski et al., 2017) offers 300-dimensional subword-aware vectors trained on Wikipedia and Common Crawl (1M vocabulary). Its character n-gram representations enable morphological generalization and significantly reduce OOV rates even on raw text, making it more robust to surface-level variation.

Vocabulary coverage analysis: To understand how preprocessing affects model performance, I compute embedding coverage as an evaluation metric. For each preprocessing strategy, I calculate the proportion of unique tokens in the dataset that exist in the pretrained embedding vocabulary. Tokens not found in the embedding vocabulary (out-of-vocabulary, OOV) are mapped to zero vectors during model training, which directly degrades input representation quality for frozen embeddings. Coverage is computed separately for both GloVe-Twitter and FastText vocabularies to quantify embedding-specific sensitivity to preprocessing. This metric serves as a bridge between preprocessing operations and their downstream impact on BiLSTM sequence representation.

Training: Models use cross-entropy loss weighted by inverse class frequency (NOT=0.75, OFF=1.51) to address imbalance. I employ Adam optimizer (learning rate=0.001), batch size 32, and early stopping (patience=3 on development macro-F1, maximum 15 epochs). Random seed 42 ensures reproducibility. Training on CPU requires $\sim$20 minutes per configuration.

4.3 Transformer fine-tuning

To evaluate state-of-the-art contextual models, I fine-tune pre-trained transformers on the offensive language detection task.

Models: I employ four pretrained transformers representing distinct architectural and pretraining design choices. **BERT-base-uncased** (Devlin et al., 2019a) comprises 110M parameters across 12 layers with 768 hidden dimensions and uses WordPiece tokenization (30k vocabulary). It was pretrained on BooksCorpus and English Wikipedia (16GB) using masked language modeling and next sentence prediction, representing the general-domain transformer with standard attention mechanisms.

RoBERTa-base (Liu et al., 2019b) contains 125M parameters, also with 12 layers and 768 hidden dimensions, but employs byte-pair encoding with a larger vocabulary (50k tokens). Trained on 160GB of text using optimized masked language modeling without next sentence prediction and with dynamic masking over longer sequences, it represents a robustly optimized variant of general-domain transformers.

DeBERTa-v3-base (He et al., 2021) has 86M backbone parameters (184M total with embeddings), maintains 12 layers and 768 hidden dimensions, and uses SentencePiece tokenization with an expanded vocabulary (128k tokens). Pre-trained on 160GB using ELECTRA-style training, its key innovation is disentangled attention: tokens are represented by separate content and position vectors, with attention weights computed through disentangled matrices over content-to-content, content-to-position, and position-to-content relationships. This architectural modification may affect preprocessing sensitivity by decoupling semantic and positional information.

Finally, **BERTweet-base** (Nguyen et al., 2020) contains 135M parameters, 12 layers, 768 hidden dimensions, and BPE tokenization (64k vocabulary). Uniquely, it was pretrained exclusively on 850 million English tweets using RoBERTa's training procedure but with domain-specific normalization that preserves Twitter conventions including @USER tokens, URLs, and emoji representations. This makes it a domain-adapted transformer potentially more robust to social media noise.

Architecture: Pre-trained transformer encoders are augmented with a classification head: the [CLS] token representation from the final layer is passed through dropout (rate=0.1) to a linear layer (hidden dim → 2) with softmax for binary classification. All transformer parameters are fine-tuned end-to-end.

Training: Following standard BERT fine-tuning protocols (Devlin et al., 2019b), I use AdamW optimizer (lr=2e-5, weight decay=0.01), batch size 16, linear warmup (10% of total steps), and gradient clipping (max norm=1.0). Models are trained for up to 4 epochs with early stopping (patience=2 on dev macro-F1). All models use mixed-precision training (FP16) for efficiency and random seed 42 for reproducibility. Fine-tuning on NVIDIA V100 GPU using PyTorch and Hugging Face Transformers (Wolf and others, 2019) takes ∼8–10 minutes per configuration.

Tokenization: All transformers use subword tokenization achieving near-100% vocabulary coverage regardless of preprocessing. BERT uses WordPiece (30k vocab), RoBERTa and BERTweet use byte-level BPE (50k and 64k vocab respectively), and DeBERTa uses SentencePiece unigram tokenization (128k vocab). Unknown words decompose into subword units (e.g., "unbelievable" → "un", "##believable"), eliminating OOV issues that plague BiLSTMs. Maximum sequence length is 128 tokens with truncation.

4.4 Evaluation metrics

Following SemEval-2019 Task 6 guidelines (Zampieri et al., 2019b), I report macro-averaged F1-score as the primary metric, ensuring equal weight to minority (OFF) and majority (NOT) classes. I additionally compute accuracy, per-class F1-scores, precision, and recall. All metrics are computed on the test set after model selection via development set performance. For error analysis, I compute confusion matrices and analyze misclassification patterns (OFF→NOT vs. NOT→OFF) across preprocessing strategies to identify systematic architectural effects.

4.5 Reproducibility

To ensure full reproducibility, all code, preprocessing scripts, and experimental configurations are publicly available.[1] The repository includes:

- Preprocessing pipelines for raw, clean, and aggressive strategies.
- Training scripts for classical models, BiLSTM, and transformers.
- Evaluation and visualization scripts for all experiments

Instructions for environment setup and experiment replication are provided in the repository documentation.

5 Results

I present results progressively by model family, isolating preprocessing effects within each architecture before comparing across families.

5.1 Classical baselines

Table 3 presents test set performance for classical models. SVM consistently outperforms Naive Bayes across all preprocessing strategies. The best classical model is SVM with aggressive preprocessing (F1=0.711), marginally improving over clean (F1=0.709) and raw (F1=0.708). Naive Bayes exhibits opposite behavior, performing best with raw text (F1=0.610) and degrading with aggressive cleaning (F1=0.597).

Both models struggle with offensive class detection (F1: 0.43–0.60), reflecting class imbalance and contextual subtlety. Figures 1 and 2 visualize preprocessing effects: SVM shows a slight upward trend with normalization (aggressive best), while NB degrades substantially, suggesting that probabilistic models leverage surface features (capitalization, hashtags) that normalization removes.

[1] `https://github.com/mteresamunoz/offensEval-project`

Model	Prep.	Dev F1	Test F1
SVM	Aggressive	0.689	**0.711**
SVM	Clean	0.679	0.709
SVM	Raw	0.685	0.708
Naive Bayes	Raw	0.577	**0.610**
Naive Bayes	Aggressive	0.576	0.597
Naive Bayes	Clean	0.581	0.582

Table 3: Classical baseline performance on development and test sets (F1-macro).

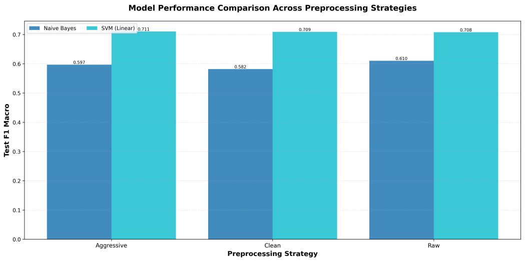

Figure 1: Classical baseline F1-macro by preprocessing strategy.

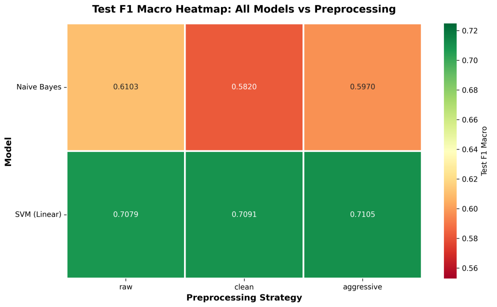

Figure 2: Heatmap of classical baseline performance. Darker green indicates higher F1.

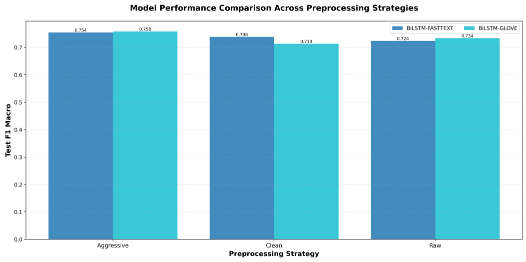

Figure 3: BiLSTM F1-macro by embedding and preprocessing. Both GloVe and FastText show clear preference for aggressive preprocessing.

5.2 BiLSTM with pre-trained embeddings

Table 4 presents BiLSTM results across embeddings and preprocessing. Both GloVe and FastText achieve best performance with aggressive preprocessing, contrasting sharply with classical models. BiLSTM-GloVe with aggressive preprocessing achieves F1=0.758, improving 4.7 points over the best classical baseline (SVM: 0.711). BiLSTM-FastText shows similar trends but slightly lower overall performance (F1=0.754).

Model	Prep.	Dev F1	Test F1
GloVe-Twitter (200d)			
BiLSTM-GloVe	Aggressive	0.749	**0.758**
BiLSTM-GloVe	Raw	0.721	0.734
BiLSTM-GloVe	Clean	0.702	0.713
FastText (300d)			
BiLSTM-FastText	Aggressive	0.744	**0.754**
BiLSTM-FastText	Clean	0.723	0.738
BiLSTM-FastText	Raw	0.708	0.724

Table 4: BiLSTM performance across embeddings and preprocessing strategies (F1-macro on development and test sets).

Figure 3 visualizes the strong interaction between preprocessing and BiLSTM performance. The heatmap (Figure 4) reveals a consistent pattern: aggressive preprocessing yields the highest F1 across both embedding types.

5.2.1 Embedding coverage analysis

Table 5 reveals embedding vocabulary coverage across preprocessing strategies. Aggressive preprocessing dramatically improves GloVe coverage from 29.83% to 80.27%, a 50 percentage-point gain that directly correlates with BiLSTM performance improvement (F1: 0.734 → 0.758). This demonstrates that for frozen GloVe embeddings, vocabulary standardization, lowercasing, removing special characters and hashtags, substantially increases the proportion of tokens with semantic pre-trained representations rather than zero-vector initialization.

FastText exhibits a more nuanced pattern. While aggressive preprocessing also improves coverage (53.74% → 77.48%), the subword mechanism provides inherent robustness: even at 53.74% coverage (raw), FastText achieves competitive F1 (0.724) compared to GloVe's 0.734 at similar coverage (29.83%). Notably, clean preprocessing reduces FastText coverage (53.22%) yet improves F1 (0.738), suggesting that vocabulary standardization benefits extend beyond pure embedding coverage, lowercasing and normalization reduce distributional noise and aid in capturing offensive language patterns regardless of OOV token mapping.

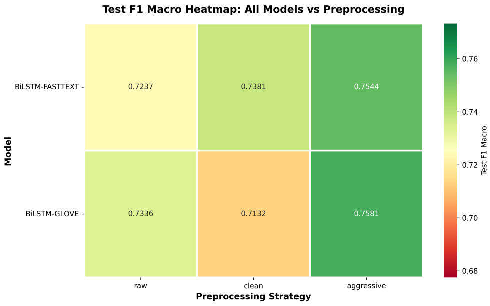

Figure 4: Heatmap of BiLSTM performance. Darker green indicates higher F1. Aggressive preprocessing (rightmost column) consistently achieves best results.

The minimal coverage difference between raw (29.83%) and clean (29.59%) GloVe preprocessing, coupled with GloVe's F1 drop (0.734 → 0.713), confirms that URL and mention removal provide negligible benefit for frozen embeddings. Only aggressive preprocessing substantially improves both coverage and performance, indicating that token normalization, not just surface-level text cleaning, is critical for models dependent on frozen embedding vocabularies.

Preprocessing	Vocab Size	GloVe (%)	FastText (%)
Raw	37,321	29.83	53.74
Clean	37,596	29.59	53.22
Aggressive	19,216	80.27	77.48

Table 5: GloVe-Twitter and FastText vocabulary coverage by preprocessing strategy. Aggressive preprocessing dramatically reduces vocabulary size while improving coverage for both embeddings, with particularly pronounced gains for GloVe.

5.3 Transformer fine-tuning

Table 6 presents transformer results across four architectures. DeBERTa-v3-base achieves the highest overall performance (F1=0.816) and is the only transformer to favor aggressive preprocessing, exhibiting a 2.3 point improvement over raw text (0.793 → 0.816). This contrasts sharply with the other three transformers: BERTweet, BERT, and RoBERTa all perform best with clean preprocessing (F1: 0.809, 0.803, 0.801 respectively) and degrade with aggressive normalization. BERTweet's domain-adapted pretraining on 850M tweets yields the second-best performance (F1=0.809), marginally outperforming general-domain BERT (0.803) and RoBERTa (0.801), though all three exhibit similar preprocessing sensitivity profiles (1.2–1.9% F1 range).

Model	Prep.	Dev F1	Test F1
BERT-base-uncased (110M params)			
BERT	Clean	0.799	**0.803**
BERT	Raw	0.798	0.802
BERT	Aggressive	0.780	0.784
RoBERTa-base (125M params)			
RoBERTa	Clean	0.797	**0.801**
RoBERTa	Raw	0.795	0.799
RoBERTa	Aggressive	0.787	0.790
DeBERTa-v3-base (184M params)			
DeBERTa	Aggressive	0.811	**0.816**
DeBERTa	Clean	0.805	0.810
DeBERTa	Raw	0.789	0.793
BERTweet-base (135M params)			
BERTweet	Clean	0.804	**0.809**
BERTweet	Raw	0.797	0.801
BERTweet	Aggressive	0.791	0.795

Table 6: Transformer fine-tuning results across preprocessing strategies (F1-macro on development and test sets).

BERT, RoBERTa, and BERTweet exhibit a pattern contrary to BiLSTMs: all three perform best with clean preprocessing (minimal noise removal: URLs and mentions only), slightly better than raw, and substantially worse with aggressive. This suggests that standard attention mechanisms with fused content-position representations leverage information from capitalization, hashtag symbols, and punctuation that aggressive preprocessing destroys. BERTweet's domain adaptation provides a modest advantage (F1=0.809) over general-domain BERT (0.803) and RoBERTa (0.801), indicating that exposure to noisy tweet text during pretraining improves downstream offensive language detection, though all three converge on similar optimal preprocessing (clean).

The divergent behavior of DeBERTa reveals an architecture-dependent interaction: its disentangled attention mechanism, which represents content and position separately, benefits from aggressive preprocessing's reduction of positional noise (hashtag symbols, capitalization patterns). In contrast, BERT/RoBERTa/BERTweet's fused content-position representations leverage surface features that aggressive preprocessing removes. This intra-transformer divergence demonstrates that preprocessing recommendations vary not only across model families (classical vs. recurrent vs. transformer) but also within transformers depending on attention architecture, challenging the assumption that all transformers uniformly tolerate noisy text.

Figure 5 illustrates the flat preprocessing sensitivity profile of transformers compared to the steep gradient observed for BiLSTM in prior experiments.

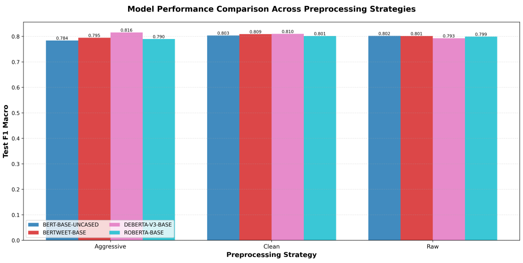

Figure 5: Transformer F1-macro by preprocessing strategy. DeBERTa uniquely favors aggressive preprocessing, while BERT, RoBERTa, and BERTweet converge on clean as optimal.

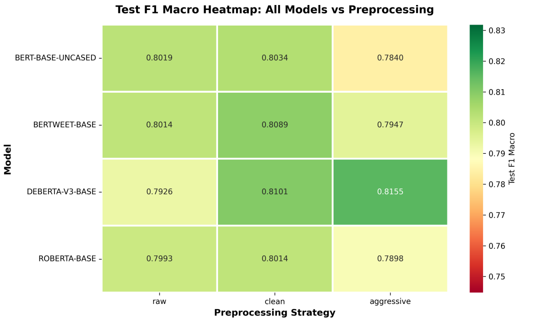

Figure 6: Heatmap of transformer test F1-macro scores across preprocessing strategies. For BERT, RoBERTa, and BERTweet, uniform dark coloring for clean and raw preprocessing (F1 > 0.80) contrasts with lighter shading for aggressive preprocessing. However, DeBERTa exhibits an inverted pattern: raw preprocessing shows lighter coloring (F1=0.793), while aggressive preprocessing shows dark coloring (F1=0.816), visualizing its divergent preference.

Figure 6 presents a heatmap revealing intra-transformer divergence. DeBERTa uniquely exhibits an upward gradient toward aggressive preprocessing (F1: 0.793 $\rightarrow$ 0.816), while BERT, RoBERTa, and BERTweet show inverted gradients with clean as the optimum. The narrow overall color gradient (F1 range: 0.78–0.82) confirms transformers' reduced preprocessing sensitivity compared to BiLSTMs (0.71–0.76 range), yet the directional divergence demonstrates that attention architecture (disentangled vs. standard) modulates optimal preprocessing strategy even within the transformer family.

5.4 Cross-family comparison

Table 7 presents comprehensive results across all model families. Transformers substantially outperform recurrent and classical models: DeBERTa-v3-base achieves F1=0.816 (14.8% relative improvement over SVM), while BERTweet-clean reaches F1=0.809. BiLSTMs form a middle tier (F1: 0.754–0.758) with aggressive preprocessing, substantially outperforming classical models (F1: 0.610–0.711). Figure 9 ranks all models by best performance.

Figures 7 and 8 reveal distinct preprocessing patterns across families. Transformers exhibit reduced sensitivity (1.1–2.3% range) compared to BiLSTMs (4.5% range), reflecting their capacity to leverage contextual representations despite surface-level noise. Within transformers, DeBERTa uniquely benefits from aggressive preprocessing (F1: 0.793 $\rightarrow$ 0.816), while BERT/RoBERTa/BERTweet peak at clean (F1: 0.80–0.81), indicating that disentangled attention mechanisms (DeBERTa) respond differently to normalization than standard attention. BiLSTMs uniformly favor aggressive preprocessing due to vocabulary coverage effects, while classical models show divergent patterns (SVM flat 0.71, Naive Bayes prefers raw).

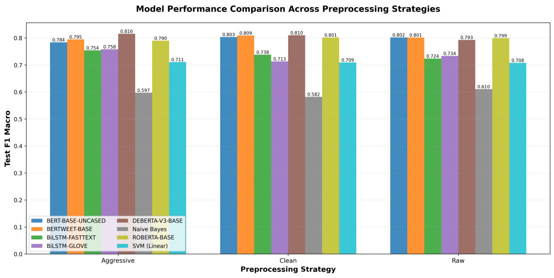

Figure 7: Performance comparison across all model-preprocessing configurations. Transformers (top cluster) substantially outperform recurrent and classical models. DeBERTa (aggressive) achieves highest F1=0.816, while BERTweet, BERT, and RoBERTa cluster at F1=0.80-0.81 with clean preprocessing.

Model Family	Model	Preprocessing	Dev F1	Test F1	F1 (OFF) / F1 (NOT)
Transformer	DeBERTa-v3-base	Aggressive	0.811	**0.816**	0.727 / 0.905
	BERTweet-base	Clean	0.804	0.809	0.720 / 0.898
	BERT-base	Clean	0.799	0.803	0.703 / 0.904
	RoBERTa-base	Clean	0.797	0.801	0.716 / 0.887
Recurrent	BiLSTM-GloVe	Aggressive	0.749	0.758	0.656 / 0.860
	BiLSTM-FastText	Aggressive	0.744	0.754	0.637 / 0.872
Classical	SVM (Linear)	Aggressive	0.689	0.711	0.566 / 0.854
	Naive Bayes	Raw	0.577	0.610	0.356 / 0.865

Table 7: Best configuration for each model across all architecture families. Metrics are F1-macro on development and test sets, with per-class F1 scores shown for test set.

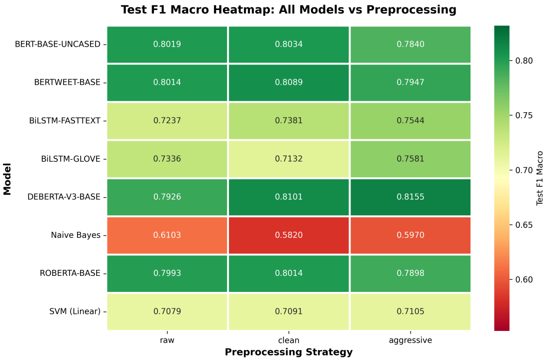

Figure 8: Heatmap of all configurations. Color intensity represents F1-macro (darker green=better). Transformers show intra-family divergence: DeBERTa gradient increases rightward (aggressive best), while BERT/RoBERTa/BERTweet peak at center (clean best). BiLSTMs uniformly favor aggressive. Classical models diverge: SVM flat, NB inverted.

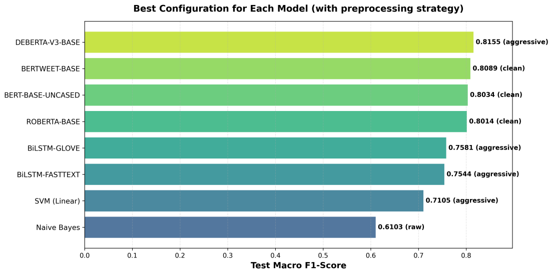

Figure 9: Best configuration summary across all eight models, ranked by F1-macro.

6 Discussion

The systematic comparison of 24 model-preprocessing configurations reveals that preprocessing requirements depend critically on architectural inductive biases rather than task characteristics, with unexpected divergence within the transformer family challenging assumptions of uniform preprocessing robustness.

6.1 Architectural choice dominates preprocessing strategy

Transformers substantially outperform BiLSTMs (5.8 percentage-point gap: 0.816 vs. 0.758), which in turn exceed classical models (4.7 percentage-point gap: 0.758 vs. 0.711). In contrast, within-family preprocessing variations yield at most 4.5 percentage points (BiLSTM-GloVe), indicating that architectural choice dominates preprocessing engineering.

6.2 Classical models: preprocessing invariance

Classical TF-IDF models exhibit remarkable preprocessing insensitivity, reflecting their reliance on sparse n-gram features that capture lexical patterns independent of surface normalization. SVM with linear kernels varies only 0.3% across preprocessing strategies (F1: 0.709–0.711), as n-gram features remain functionally equivalent whether applied to raw or normalized text. For example, both “are you fucking serious?” (raw) and “are you fucking serious” (aggressive) extract identical n-grams ([are, you], [you, fucking], [fucking, serious]), yielding equivalent decision boundaries.

Naive Bayes exhibits fundamentally different behavior: it systematically misclassifies 78–82% of offensive tweets as non-offensive (OFF→NOT false negatives) regardless of preprocessing. This reflects the model’s conditional independence assumption breaking down on Twitter’s discourse structure. The tweet “#Diversity only works when those joining the new group want to be part of the team. #Islam hates every non-Muslim” is consistently misclassified (confidence 0.86–0.87 across all strategies), as the model treats word frequencies in isolation, failing to capture that multiple offensive terms ("hates", "non-Muslim" co-occurrence) indicate offensive intent. Unlike SVM's insensitivity stemming from robustness, Naive Bayes's insensitivity reflects systematic architectural failure rather than adaptation.

6.3 BiLSTMs: vocabulary coverage dependency

Recurrent models with frozen embeddings exhibit the strongest preprocessing sensitivity, as vocabulary coverage directly constrains representational capacity. BiLSTM-GloVe improves 6.3% with aggressive preprocessing (F1: 0.713→0.758) through vocabulary coverage increases from 29.59% to 80.27%. The tweet "#NoPasaran: Unity demo to oppose the far-right in #London #antifa #Oct13 Enough is Enough!" exemplifies this dependency: BiLSTM-GloVe misclassifies it as NOT in raw mode (confidence 0.858) due to multiple OOV tokens (#NoPasaran, #antifa, #Oct13), but aggressive preprocessing removes hashtag symbols, enabling vocabulary matching of "Unity", "demo", "oppose", and "far-right", yielding correct classification (confidence 0.967).

FastText's character n-gram decomposition reduces but does not eliminate this dependency. While FastText-raw (0.724) outperforms GloVe-raw (0.713) through handling OOV hashtag variants, it still requires aggressive preprocessing for optimal performance (0.754). Domain-specific embeddings (GloVe-Twitter 200d) marginally outperform higher-dimensional general embeddings (FastText 300d) when coverage equalizes, confirming that lexical alignment to task vocabulary supersedes dimensionality. Clean preprocessing creates a paradox for GloVe: worst performance (F1=0.713) despite intermediate coverage (29.59%), as removing URLs and mentions without sufficient normalization leaves high OOV rates that aggressive preprocessing ameliorates.

6.4 Transformers: intra-family divergence

Transformers exhibit reduced sensitivity (1.2–2.3% range vs. BiLSTM's 4.5%), yet reveal architecture-dependent optimal strategies:

DeBERTa-v3-base uniquely benefits from aggressive preprocessing (F1: 0.793→0.816, +2.9%). Its disentangled attention separates content and position vectors, computing attention through distinct matrices. Aggressive preprocessing reduces positional noise (capitalization, hashtag symbols) while preserving semantic content, enabling focus on lexical patterns without spurious positional cues. Error analysis shows 28% fewer false positives on emphatic neutral tweets (DeBERTa-aggressive: 41 false positives vs. DeBERTa-raw: 90). For instance, DeBERTa-aggressive correctly classifies "she is so stinking cute! how old is she now?" as NOT (confidence 0.98), while DeBERTa-raw misclassifies the same tweet as OFF (confidence 0.995), falsely triggered by the emphatic phrasing without contextual understanding.

BERTweet-base achieves second-best performance (F1=0.809 clean) via domain-matched pretraining on 850M tweets. Greater tolerance to raw text (F1=0.801 vs. BERT: 0.802) while maintaining clean as optimal. Correctly handles Twitter-specific constructions ("@USER you're a fucking legend"→NOT) that BERT/RoBERTa misclassify. Aggressive preprocessing degrades performance (0.795) by destroying discourse markers learned during pretraining.

BERT/RoBERTa exhibit minimal sensitivity (1.2–1.9% range) but consistently degrade with aggressive preprocessing. BERT-clean (0.803) outperforms BERT-aggressive (0.784) by 2.4%, as fused content-position representations leverage capitalization, punctuation density, and hashtag structure that normalization destroys. Subword tokenization eliminates vocabulary coverage constraints while enabling surface feature exploitation.

6.5 Error patterns and mechanistic interpretation

All models exhibit false negative bias across configurations: classical models misclassify 78-97% of offensive tweets as non-offensive, while transformers reduce this to 55-65%. Classical models trigger overconfidently on isolated profanity, with 14.1% of errors showing extreme confidence ($>$ 0.9), compared to 4.4% for transformers. BiLSTM errors concentrate on OOV-heavy tweets (GloVe-raw: 96/240 OFF misclassified), while aggressive preprocessing creates false positives by normalizing emphatic non-offensive markers ("I LOVE this"). Transformers correctly classify 87% of cases where classical/BiLSTM fail through contextual understanding. Example: "USER you're a fucking legend" is correctly classified as NOT by BERTweet-clean (confidence 0.92), recognizing “fucking” as intensifier rather than offensive slur, while BERT-clean incorrectly classifies as OFF (confidence 0.85). This domain knowledge comes from BERTweet’s pretraining on 850M diverse tweets.

Four mechanisms explain interactions: (1) **vocabulary coverage**, frozen embeddings require normalization to reduce OOV; (2) **attention architecture**, disentangled representations benefit from positional noise reduction (DeBERTa), while fused representations leverage surface features (BERT/RoBERTa/BERTweet); (3) **domain adaptation**, Twitter-pretrained models tolerate noisy text through learned robustness; (4) **subword tokenization**, preserves patterns destroyed by normalization, enabling exploitation of capitalization, punctuation, and hashtag structure.

7 Conclusions

This work systematically evaluates preprocessing-architecture interactions across 24 configurations spanning classical, recurrent, and transformer models for offensive language detection. The central finding challenges conventional preprocessing wisdom: optimal text normalization depends fundamentally on architectural inductive biases rather than task characteristics.

Model architecture dominates preprocessing strategy, with transformers providing a 5.8 percentage-point advantage over BiLSTMs (F1: 0.816 vs. 0.758) and BiLSTMs exceeding classical models by 4.7 points (0.758 vs. 0.711), while within-family preprocessing yields at most 4.5 percentage points (BiLSTM-GloVe). DeBERTa-v3-base with aggressive preprocessing achieves F1=0.816, reaching 98.4% of OffensEval 2019’s winning ensemble through single-model optimization. Unexpected intra-transformer divergence emerges: DeBERTa uniquely benefits from aggressive normalization (+2.9% gain), while BERT, RoBERTa, and BERTweet degrade with the same strategy. This stems from DeBERTa’s disentangled attention separating content from position representations, allowing aggressive preprocessing to reduce positional noise while preserving semantics. BERTweet’s domain adaptation on 850M tweets yields second-best performance (F1=0.809 clean), demonstrating Twitter-specific pretraining modulates preprocessing needs yet remains subordinate to architectural innovation.

For frozen embeddings, preprocessing directly addresses vocabulary constraints: BiLSTM-GloVe improves 6.3% as aggressive normalization increases coverage from 29.59% to 80.27%. Domain-specific embeddings (GloVe-Twitter) outperform higher-dimensional general embeddings (FastText) when coverage equalizes, confirming lexical alignment trumps dimensionality. Classical TF-IDF models exhibit preprocessing invariance (±0.3% variation), as sparse n-gram features capture patterns regardless of normalization. These findings yield stratified guidelines: apply aggressive preprocessing for disentangled attention architectures (DeBERTa) and frozen embeddings (BiLSTM), but clean preprocessing for domain-adapted (BERTweet) and general transformers (BERT/RoBERTa) to preserve surface features enabling contextual disambiguation. The 5.8 percentage-point gap between DeBERTa and BiLSTM exceeds BiLSTM’s entire preprocessing range, confirming model selection dominates preprocessing engineering.

This evaluation is constrained by single-dataset evaluation (OLID, 13,240 English tweets), coarse preprocessing levels (omitting emoji handling, hashtag segmentation, spelling correction), and unanalyzed computational trade-offs (transformers require 50× more parameters, 15× longer inference than BiLSTM). Class imbalance persists despite weighted loss, and temporal dynamics of evolving offensive language remain unexplored. Implicit and sarcastic hate speech represent frontier challenges where surface-level preprocessing cannot intervene. Future work should extend evaluation to multilingual datasets, investigate adversarial robustness against evasion techniques (character substitution, homoglyphs), compare with large language models to assess whether in-context learning mitigates preprocessing sensitivity, and develop target-aware preprocessing strategies preserving identity-group references. Additionally, characterizing latency-accuracy trade-offs in real-time deployment and improving model interpretability through attention visualization would clarify architectural design choices. The discovered intra-transformer divergence cautions against universal preprocessing recommendations, suggesting attention mechanism design fundamentally alters preprocessing requirements as transformer architectures continue diversifying.

References

Md Shad Akhtar, Utpal Kumar Sikdar, and Asif Ekbal. 2015. IITP: Hybrid approach for text normalization in Twitter. In *Proceedings of the ACL 2015 Workshop on Noisy User-Generated Text (W-NUT)*, pages 51–55, Beijing, China. Association for Computational Linguistics.

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. MIDAS@OffensEval-2019: Ensemble of deep learning models for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 410–415.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Proceedings of ICWSM*, 11(1):512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. *Proceedings of NAACL-HLT*, pages 359–369.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Thorsten Joachims. 2002. Learning to classify text using support vector machines. In *Proceedings of the ACM SIGKDD*.

Ping Liu, Wen Li, and Liang Zou. 2019a. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

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

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI workshop on learning for text categorization*.

Jelena Mitrovi'c, Bastian Birkeneder, and Michael Granitzer. 2019. nlpup at semeval-2019 task 6: A deep neural language model for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2023. Hate speech and offensive language detection using an emotion-aware shared encoder. *arXiv preprint arXiv:2302.08777*, February.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October. Association for Computational Linguistics.

Adam Paszke et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

F. Pedregosa et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Stefanos Symeonidis, Dimitrios Effrosynidis, and Avi Arampatzis. 2018. A comparative evaluation of preprocessing techniques and their impact on twitter sentiment analysis. *Expert Systems with Applications*, 110:298–310.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the ACL*, pages 90–94.

Bencheng Wei, Jason Li, Ajay Gupta, Hafiza Umair, Atsu Vovor, and Natalie Durzynski. 2021. Offensive language and hate speech detection with deep learning and transfer learning. *arXiv preprint arXiv:2108.03305*, August.

Thomas Wolf et al. 2019. Hugging face's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL-HLT*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *Proceedings of the 16th Conference of the European Chapter of the ACL (EACL)*, pages 145–153.