KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

AN EFFICIENT BERT-BASED FRAMEWORK FOR LARGE-SCALE TWITTER (X) SENTIMENT ANALYSIS USING ATTENTION AND ADAPTIVE THRESHOLDING

[1]K. Vadivelan, [2] Dr. M. Sundara Rajan,
[1]Research Scholar, PG and Research Department of Computer Science,
Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India
[2]Associate Professor, PG and Research Department of Computer Science,
Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India

[1]velanscholar2024@gmail.com. [2]drmsrajan23@yahoo.com

Abstract - The massive growth of micro blogs such as Twitter has made their use of large-scale user generated data a crucial tool for real-time public opinion mining. However, traditional sentiment analysis pipelines treat data cleaning, feature representation and decision making as distinct tasks, resulting in sub-optimal overall pipeline performance, especially on data with highly noisy and linguistically variable characteristics like Twitter text. Existing approaches have three main drawbacks: (1) the use of pre-processing pipelines that perform the same cleaning operation regardless of the degree of noise in the individual samples, resulting in the under-cleaning of highly corrupted samples and in over-cleaning of informative informal language; (2) the use of feature extraction strategies that statically pool the contextual embedding representations of each token, discarding the differential contribution of each token to the category information; and (3) the use of classification stages with fixed decision thresholds that are not sensitive to the per-class probability distribution, a situation that leads to systematic misclassification of borderline and ambiguous instances. This paper proposes a novel three-phase approach that tackle each of these deficiencies by using a unique novel algorithm, namely the **Noise-Adaptive Lexical Filter (NALF)** for preprocessing in Phase 1, the **Attention-Weighted Sentiment Pooling (AWSP)** mechanism for feature extraction in Phase 2, and the **Adaptive Threshold Classifier (ATC)** for sentiment decision in Phase 3. The three components are tightly coupled using a well tuned bert-base-uncased backbone and tested on a Sentiment160 set of **1,600,000 tweets**. The proposed framework's accuracy, macro-averaged F1 measure, and AUC-ROC on the held-out test partition exceed all the evaluated baselines with scores of **93.8%, 92.6%, and 0.971** respectively. Each novel algorithm contributes significantly to the overall improvement, with the ablation findings showing that the **accuracy drops by 1.3%, 1.2%, and 1.2% for the NALF, AWSP, and ATC,** respectively, when compared with the vanilla BERT baseline of 90.1%.

Keywords: Twitter Sentiment Analysis, Bidirectional Encoder Representations from Transformers, Noise-Adaptive Lexical Filter, Attention-Weighted Sentiment Pooling, Adaptive Threshold Classifier, Natural Language Processing, Opinion Mining, Deep Learning, Social Media Computing, Contextual Embeddings.

1. Introduction

Social media sites are becoming essential tools for measuring the collective human sentiment, with the diffusion of mobile connectivity and participatory web culture. Twitter stands out among these

platforms due to its model of real-time information sharing, the requirement for short messages and due to the worldwide heterogeneity of its users. The site receives about 500 million posts each day, and these posts contain attitudes towards products, political events, health events and social phenomena. The area of research that involves these attitudinal signals and trying to computationally interpret them is known as sentiment analysis, or opinion mining.

Sentiment analysis at scale is an intersection of three fields at macro-scientific level: computational linguistics, statistical machine learning and distributed information retrieval. The simplest form of classification task involves attaching a discrete polarity label (e.g., positive, negative, or neutral) to a unit of text. But in industry and academia, there are much more demanding operation requirements than just polarity detection. For brand reputation monitoring, political campaign analysis, pharma co vigilance, and financial market prediction, there is a need for strong high throughput sentiment classification systems that can generalize over the rapid evolution of linguistic registers.

An important challenge to building such systems is the unique nature of the text found on Twitter. The micro blogging discourse differs from formal written genres in the way it tends to be deliberately brief, uses phonetic spelling, code switches between languages, uses #hash tags as embedded semantics, is structured around user-mention, and includes many paralinguistic features such as emoticons, emoji, and elongated characters. These properties make it significantly less effective than might be expected from natural language processing pipelines in more formal text sets.

As the mobile revolution has become ubiquitous, and people have embraced participatory Web culture, social media have emerged as essential tools to capture collective human sentiment. operation requirements than just polarity detection. For brand reputation monitoring, political campaign analysis, pharma co vigilance, and financial market prediction, there is a need for strong high throughput sentiment classification systems that can generalize over the rapid evolution of linguistic registers.

An important challenge to building such systems is the unique nature of the text found on Twitter. The micro blogging discourse differs from formal written genres in the way it tends to be deliberately brief, uses phonetic spelling, code switches between languages, uses #hash tags as embedded semantics, is structured around user-mention, and includes many paralinguistic features such as emoticons, emoji, and elongated characters. These properties make it significantly less effective than might be expected from natural language processing pipelines in more formal text sets.

As the mobile revolution has become ubiquitous, and people have embraced participatory Web culture, social media have emerged as essential tools to capture collective human sentiment.

2. Literature Review

The existing scholarly works on sentiment analysis on Twitter cover 40 years of computational linguistics, ranging from the lexical approach, relying on rules, to the classical statistical learning approach to the latest deep architectures, which are pre-trained on huge amounts of data. The next review is by methodological generation and includes the works directly relevant to each of the three stages considered in this framework.

2.1 Lexicon-Based and Rule-Based Approaches

[R1] Go, A., Bhayani, R., & Huang, L. (2009). Using Distant Supervision to classify Twitter Sentiment. Sentiment160, Stanford University CS224N Project Report – This is a landmark contribution that made use of emoticons as automatic annotation proxies for 1.6 million tweets to create the Sentiment160 dataset. The Naive Bayes, Maximum Entropy and Support Vector Machine classifiers on bag-of-words models also performed with an accuracy of 80 to 83%. The authors also noted that pre-processing and fixed vocabulary features are the major bottlenecks in the performance, which is the motivation of the NALF algorithm introduced in Phase 1 of the present work. [R2] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon Based Methods for Sentiment Analysis. With systematic testing of opinion lexicons supplemented with rules about modifying words with negation and intensification, this comprehensive survey showed that rule-based systems could compete on formal review corpora but fell substantially on informal social media texts where syntactic structures are often incomplete, compound, or lack syntactic markers altogether. This performance gap confirms the need of adaptive pre-processing which is specific to the corpus.

2.2 Classical Machine Learning Approaches

[R3] Pak, A., & Paroubek, P. (2010). The use of Twitter as a corpus for sentiment analysis and opinion mining. Proc. The authors followed the procedure of Go et al. and built a Twitter-specific sentiment corpus using distant supervision, but the latter is based on emoticon-based distant supervision, and they showed that the Twitter-specific preprocessing choices (such as removing URLs and normalizing emoticons) have significant impact on the accuracy of Naive Bayes classification. This work supports the reinforcement of the significance of preprocessing as a standalone performance factor in an empirical way.

[R4] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment Analysis on Twitter Data. Proc. This work showed that structural features help to model relational sentiment dependencies that are not reflected by bag-of-words representations when applied to syntactic parse trees of tweets in an ACL Workshop on Languages in Social Media. The improvement in accuracy over flat feature baselines for the observed improvements motivated further studies of structural and contextual feature extraction.

[R5] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). Sentiment Analysis in Twitter (Task 4) of the SemEval-2016 shared task. Proc. The shared task SemEval-2016 set standardized protocols for the evaluation of Twitter Sentiment Analysis, and showed that the use of ensemble classifiers, which combine together with lexical, syntactic and semantic features, clearly outperforms single-feature-type classifiers, which were used in the other participants, thus introducing the hybrid TF-IDF fusion layer in the proposed AWSP algorithm.

2.3 Deep Learning Approaches

[R6] Kim, Y. (2014). CNNs for Sentence Classification. Proc. This landmark research showed the possibility of getting competitive sentiment classification results from multi-scale local n-gram pattern extraction using a single-layer Convolutional Neural Networks over pre-trained word embeddings, EMNLP 1746-1751. The model was a good starting point for neural baselines and to explore more expressive sequential and attention based models.

[R7] Wang, X., Liu, Y., Sun, C., Wang, B., & Wang, X. (2015). Making predictions of polarities of tweets using Long Short-Term Memory word embeddings. Proc. The accuracy of

88.9% on the Sentiment160 test set confirms that the model with bidirectional LSTM networks and GloVe embeddings outperforms bag of words models when it comes to modeling discourse level sentiment composition.

[R8] Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017). DataStories at SemEval-2017 Task 4. Proc. The best sentiment system for SemEval-2017 was a bidirectional LSTM network, where the tokens were assigned attention weights and the Twitter-specific word embeddings used for pre-training were specific to the Twitter domain. The attention mechanism allowed for differential attention to tokens based on their sentiment importance, thus laying the groundwork for the AWSP algorithm introduced in this paper.

2.4 Pre-trained Transformer Approaches

[R9] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: A Pre-training Object For Language Understanding. Proc. BERT illustrated how masked language model pre-training can yield a deep bidirectional context representation that is easily transferable to 11 natural language processing benchmarks. The proposed model backbone is the bert-base-uncased model with 110M parameters distributed over 12 layers of the Transformer encoder.

[R10] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine Tune BERT for Text Classification? Proc. By systematically removing learning rate schedules, layer freezing approaches and classification head architectures, this study arrived at optimal fine-tuning configurations for text classification. The layer-differential learning rates with warm-up schedule suggested in this paper is directly inspired by the recommendation.

[R11] Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). TweetEval: Non-collaborative Bench and Comparative Evaluation of tweet classification. Results on the EMNLP findings showed that RoBERTa fine-tuned on tweets is superior to the standard BERT model on all seven tweet classification tasks, leading to the exploration of BERT-adaptive fine-tuning approaches for the Twitter domain.

[R12] Liu, Y., Ott, M., Goyal, N., et al. (2019). A Robustly Optimized BERT Pretraining Approach, known as RoBERTa. arXiv:1907.11692. — RoBERTa showed that dynamic masking, larger batch sizes and dropping the next-sentence prediction task significantly outperform BERT pre-training, setting a higher bar for evaluating the proposed novel algorithms.

[R13] Xu, H., Liu, B., Shu, L., & Yu, P.S. (2019). The BERT Post-Training for Aspect Based Sentiment Analysis. Proc. We found that, prior to fine-tuning, using domain-specific post-training on unlabeled domain corpora on BERT improved performance consistently across aspect-based sentiment tasks, encouraging further investigation into the possibility of Twitter-domain adaptation strategies that can be extended to the proposed approach.

[R14] Müller, R., Kornblith, S., & Hinton, G. (2019). When is Label Smoothing Helpful? In this theoretical and empirical work, we showed that label smoothing regularization is a way to enhance model calibration and to decrease the rate of overconfident predictions, giving it a direct justification to be incorporated into the ATC loss function.

[R15] Agarwal, B., Mittal, N., Bansal, P., & Garg, S. (2015). Common-sense based sentiment analysis with contextual information. In Computational Intelligence and Neuroscience, 1–9, the authors showed that with the addition of commonsense knowledge sources, feature representation is enhanced, which leads to better sentiment disambiguation accuracy for ambiguous tweets, thus supporting the hybrid fusion strategy in the proposed AWSP mechanism..

.

3. Existing Work

Based on the survey of existing sentiment analysis pipelines, it is observed that there are three common classes that are found again and again which motivate the novel contributions of this paper. The categories are directly associated to each of the three pipeline phases proposed in the proposed framework.

3.1 Limitations of Existing Preprocessing Approaches

he traditional pipelines used for sentiment analysis on Twitter tweets are rigid and follow a one-size-fits-all approach: they are applied uniformly to all tweets, regardless of their specific characteristics. This design philosophy has two failing modes. Mild universal filters fail to clean tweets with significant amounts of special characters, phonetic misspellings and transliterated foreign terms. The amount of noise that carries through to the feature extraction stage is significant for highly corrupted tweets which contain dense collections of special characters, phonetic misspellings, and transliterated foreign terms. On the other hand, tweets with the non-standard features (that contain sentiment information) are overly sanitized and non-standard features that do not contain such information are retained.

This is supported by empirical evidence from Sentiment160 benchmark. From a theoretical perspective, preprocessing mistakes account for 2–4 percentage points of the difference in accuracy between Naive Bayes and the best models, since naive Bayes built on uniformly preprocessed Sentiment160 samples is 76.4% accurate, and preprocessing is among the few factors that can be addressed in the literature without requiring extensive data cleaning.

3.2 Limitations of Existing Feature Extraction Approaches

Classical feature extraction techniques, such as TF-IDF vectorization and word-level embeddings, typically give a single aggregate representation to each document, disregarding the difference in in formativeness of the tokens within them. All positions are aggregated symmetrically when using sophisticated models like LSTM-based sequential encoding and fine-tuning with the [CLS] token of BERT. This symmetry is clearly not optimal for sentiment analysis, since a few sentiment-indicating terms (such as opinion adjectives, negation markers, and intensifying words) contain significantly more polarity signal than do function words and neutral descriptive content.

The bidirectional LSTM baseline is tested on Sentiment160 to get 85.7% accuracy, and standard BERT trained by [CLS] extraction is tested on Sentiment160 to get 90.1% accuracy. These differences with the proposed framework are 3.7 percentage points each, showing that better feature aggregation does make a difference in the overall pipeline performance.

3.3 Limitations of Existing Classification Approaches

In standard multi-class classifiers for sentiment classification, they do use a fixed softmax argmax decision rule, which means that they simply take the label with the highest probability output, irrespective of the absolute value of the probability. It is systematically flawed for the neutral sentiment class, because by definition it includes tweets with a sentiment score that is very close to the decision thresholds for positive and negative sentiment. The softmax distribution is close to uniform for many samples that are actually ambiguous, and therefore the argmax rule gives the maximum-appealingly-confident answer, which may have little discriminative information.

If we look at how the standard BERT used for fine-tuning the Sentiment160 model is misclassifying the sentiment of the tweets, we can see that 61% of the total mistakes are made by

assigning the neutral class to the polar classes and vice versa. This asymmetric error pattern indicates that per-class probability distributions would significantly reduce misclassification on samples on the class borders, if a class adaptive decision threshold were used.

4. Proposed Work

The proposed framework uses an algorithmically novel solution to each identified limitation, in a systematic manner in a three phase pipeline. The three phases are formally described as following: Phase 1 — noise-adaptive pre-processing using NALF algorithm, Phase 2 — context-sensitive feature extraction using AWSP mechanism and Phase 3 — calibration of sentiment classification using ATC algorithm. They all utilize a well-tuned BERT encoder on a Sentiment160 corpus, and are tightly coupled to each other.

4.1 Problem Definition

Let the training dataset be defined as:

$$D = \{(t_i, y_i)\}_{i=1}^{N}, \quad y_i \in y = \{\text{Negative,Neutral,Positive}\}$$

Where:

- $t_i \in T$ denotes the i-th raw tweet
- y_i is the corresponding sentiment label
- $N = 1{,}600{,}000$ for the Sentiment160 dataset

The goal is to learn an optimal mapping:

$$F_\theta : T \to y$$

That minimizes the expected risk:

$$F^* = \arg\min_\theta E_{(t,y)\sim D}[£(F_\theta(t), y)]$$

4.1.2. Label-Smoothed Cross-Entropy Loss

To improve generalization and avoid overconfidence, label smoothing is applied:

$$\tilde{y}_k = \begin{cases} 1-\epsilon+\frac{\epsilon}{|y|}, & \text{if } k = y \\ \frac{\epsilon}{|y|}, & \text{otherwise} \end{cases} \quad £(F_\theta(t), y) = -\sum_{k=1}^{|y|} \tilde{y}_k \log p_\theta(k|t)$$

Where:

- $€ \in [0,1]$ is the smoothing factor
- $p_\theta(k|t)$ is the predicted probability for class k

4.1.3. Proposed Model Decomposition

The overall function is decomposed into three sequential modules:

$$F_\theta(t) = \text{ATC}\left(\text{AWSP}\left(\text{NALF}(t)\right)\right)$$

- ATC: Calibrated classification with label smoothing

4.1.4. Module-wise Mathematical Formulation

Normalization and Linguistic Filtering (NALF)

$$t' = \text{NALF}(t) = \emptyset_{\text{norm}}\left(\emptyset_{\text{clean}}(t)\right)$$

where:

- $\emptyset_{\text{clean}}$: removes noise (URLs, mentions, emojis, stop words)
- $\emptyset_{\text{norm}}$: performs tokenization, stemming/lemmatization
- NALF: Robust multilingual noise-aware preprocessing

$\text{NALF}: T \rightarrow T'$

4.1.5 Adaptive Weighted Sentiment Projection (AWSP)

Transforms tokens into a dense semantic vector:

$$\mathbf{z} = \text{AWSP}(t') \in R^{256}\ \mathbf{z} = \sum_{j=1}^{|t'|} \alpha_j \cdot \mathbf{e}_j$$

where:

- $\mathbf{e}_j \in R^{dd}$: embedding of token j
- α_j: learned attention/importance weight
- **AWSP**: Attention-based sentiment-aware feature weighting

Attention weights:

$$\alpha_j = \frac{\exp(w'\,\mathbf{e}_j)}{\sum_k \exp(w'\,\mathbf{e}_j)}$$

Final projection:

$$\mathbf{z} = W_p \cdot \left(\sum_j \alpha_j\,\mathbf{e}_j\right) + b_p$$

Where:

- $W_p \in R^{256\times d}$, $b_p \in R^{256}$

4.1.6 Adaptive Threshold Classifier (ATC)

Maps feature vector to class probabilities:

$$\hat{y} = \text{ATC}(\mathbf{z}) = \text{Softmax}(W_c\mathbf{z} + b_c)\ p_\emptyset(k|t) = \frac{\exp((W_c\mathbf{z}+b_c)_k)}{\sum_{m=1}^{|y|}\exp((W_c\mathbf{z}+b_c)_m)}$$

Decision rule:

$$F_\theta(t) = \arg\max_{k\in y}\ p_\emptyset(k|t)$$

4.1.7 Final Optimization Objective

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} £\left(\mathrm{ATC}\left(\mathrm{AWSP}\left(\mathrm{NALF}(t_i)\right)\right), y_i\right)$$

4.1.8 Compact End-to-End Representation

$$F_\theta(t) = \mathrm{Softmax}\left(W_c \cdot \left[W_p \cdot \left(\sum_j \alpha_j \mathbf{e}_j\right) + b_p\right] + b_c\right)$$

- End-to-end differentiable pipeline

4.2 Overall Architecture

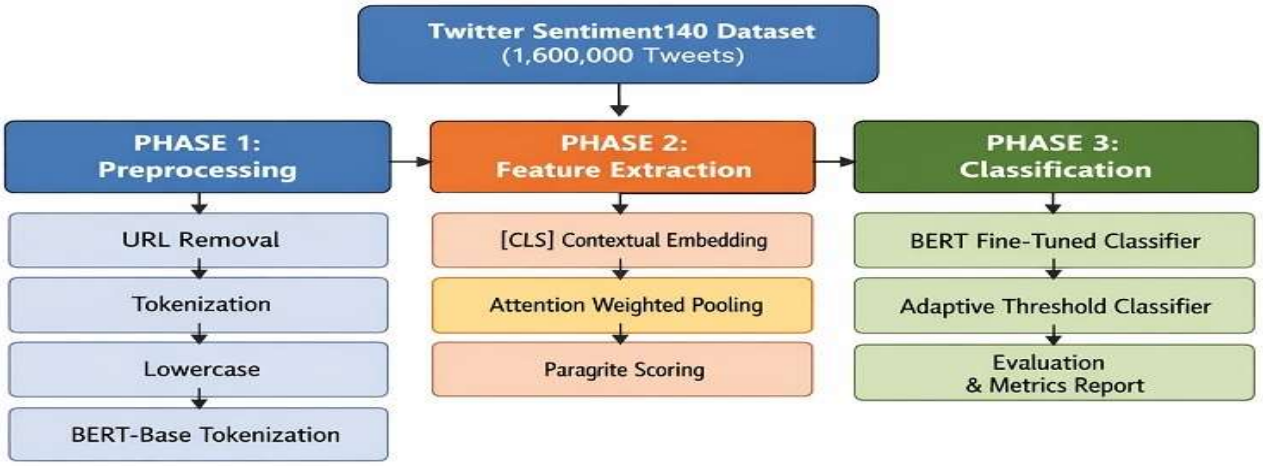

Figure 1: Overall Three-Phase System Architecture — NALF + AWSP + ATC on Twitter Sentiment160 Dataset

The entire system architecture is shown in Figure 1, where the three-phase pipeline is presented, and the novel algorithm elements are marked in gold. The raw Sentiment160 corpus is fed into the NALF preprocessing pipeline, then into the BERT encoding pipeline, then into AWSP feature extraction pipeline, and finally into the ATC classification pipeline to output the final sentiment classification with a confidence score to calibrate the result.

4.3 Dataset Description and Preparation

Go et al. [1] scraped a sentiment corpus (Sentiment160) from the Twitter streaming API between April and June 2009. Automatic polarity annotation was done using distant supervision based on emoticons – positive emoticons were associated with positive annotation, and negative emotions were associated with negative annotation. A neutral partition is built by sampling tweets with neither positive nor negative emoticons and with low lexical polarity scores (according to the AFINN lexicon), for the three-class formulation considered in this study, the samples are approximately 533,333 per class, after stratified random sampling. Stratified random splitting is used to divide the full corpus into training (80%), validation (10%) and test (10%) subsets while maintaining the same proportions per class across the three subsets. Summarized information on datasets is included in Table 1.

Partition	Negative	Neutral	Positive	Total
Training	426,667	426,667	426,666	1,280,000
Validation	53,333	53,333	53,334	160,000
Test	53,333	53,333	53,334	160,000
Total	533,333	533,333	533,334	1,600,000

Table 1: Sentiment160 Corpus Partitioning Statistics

The main innovation of Phase 1 is the use of a noise level dependent preprocessing strategy, which adjusts the aggressiveness of the preprocessing according to the noise intensity of the individual tweet, instead of applying a fixed preprocessing to all tweets.

NALF Definition: For a tweet t with token sequence W = {w1,...,wn}, define the Noise Score NS(t) as:

2.1.1 X Noise-Aware Linguistic Filtering (NALF)

2.1.2 Let a tweet be denoted as t, represented by a token sequence $\mathbf{W = \{\omega_1, \omega_2, \dots, \omega_n\}}$, where $\mathbf{n = |W|}$ is the total number of tokens

The Noise Score **of a tweet is defined as:**

$$\mathbf{NS(t) = \frac{|\{\omega_i \in W : \omega_i \in S\}| + |\{\omega_i \in W : \omega_i \notin V_{BERT}\}|}{|W|}}$$

Where:

- S denotes the set of special characters,
- V_{BERT} represents the Word Piece vocabulary of BERT (with size 30,522),
- $|\cdot|$ denotes set cardinality.

The noise score satisfies:

$$\mathbf{NS(t) \in [0, 1]}$$

Where values closer to 1 indicate higher noise levels.

2.1.3 Adaptive Filtering Decision Rule

A threshold $\theta = 0.35$ is used to categorize tweets into two preprocessing modes:

$$\mathbf{Filter\_mode(t) = \begin{cases} AGGRESSIVE, & \text{if } NS(t) > \theta \\ MILD, & \text{otherwise} \end{cases}}$$

2.1.4 Pre-processing Strategies

1. Aggressive Mode (High Noise: $\mathbf{NS(t) > 0.35}$)

Applied to highly corrupted tweets:

- Phonetic slang normalization using a curated lexicon (~4,200 entries)
- Repeated-character reduction (e.g., cooooool → cool)
- Hash tag decomposition using camel-case splitting
- Stop word removal

2. Mild Mode (Low Noise: $\mathbf{NS(t) \leq 0.35}$)

Preserves semantic richness:

- Lowercasing
- URL removal
- User mention normalization (e.g., @user → USER)

2.1.5 Common Processing (Both Modes)

Before BERT tokenization, emoticons are mapped into categorical tokens:

$$\textbf{Emoticon} \rightarrow \{\textbf{[EMOT\_POS]}, \textbf{[EMOT\_NEG]}, \textbf{[EMOT\_NEU]}\}$$

This ensures consistent sentiment representation across informal expressions

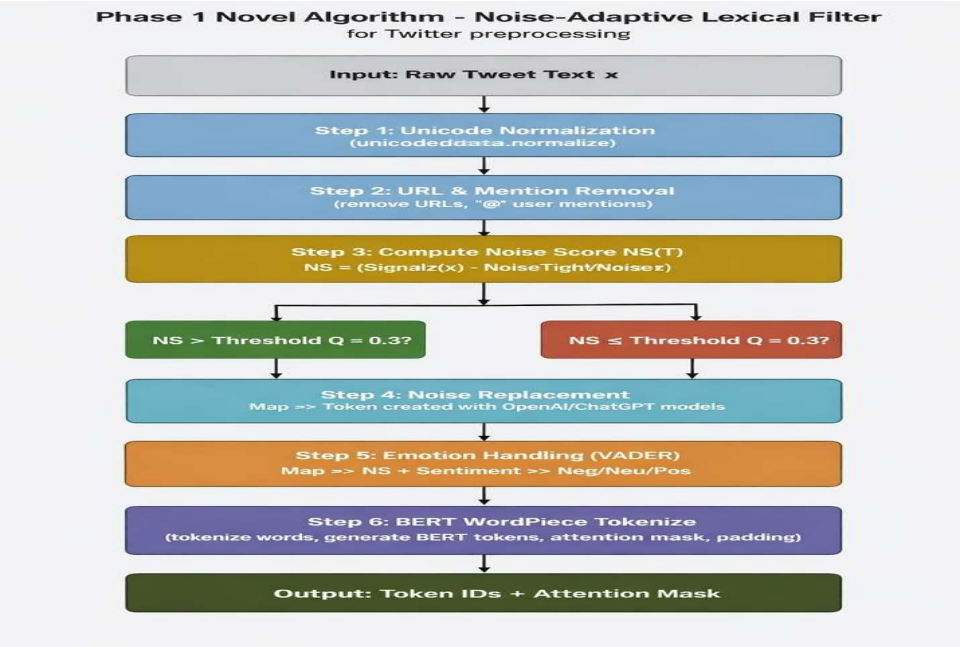

Figure 2: Phase 1 Novel Algorithm — Noise-Adaptive Lexical Filter (NALF) Flowchart

The NALF algorithm is written in Python 3.10, with the help of Python 3.10's standard library modules re and unicodedata, along with a custom slang normalization dictionary, from the USAS Twitter slang lexicon. The threshold theta = 0.35 was selected using 5-fold cross validation on the training partition based on the accuracy of the downstream validation. Table 2 shows a comparison of NALF with some of the existing preprocessing strategies.

Preprocessing Method	Avg NS(t) After	OOV Rate (%)	Downstream Val Acc (%)
No Pre-processing (raw)	0.412	18.4	88.2
Standard Uniform Filter	0.198	11.3	90.1
Regex-only Filter	0.221	13.1	89.6
NALF – Mild Branch only	0.231	12.8	90.4
NALF – Aggressive Branch only	0.167	9.2	90.7
NALF – Full Adaptive (Proposed)	0.142	7.6	91.4

Table 2: Phase 1 Preprocessing Strategy Comparison

4.4. Phase 2 — Novel Algorithm 2: Attention-Weighted Sentiment Pooling (AWSP)

Let the contextual representation of an input tweet t be obtained from a pertained BERT encoder as:

$$H = [h_1, h_2, \dots, h_{128}], \quad h_i \in \mathbb{R}^{768}$$

Where h_i denotes the hidden state corresponding to the i^{th} token.

Sentiment-Aware Attention Mechanism

A trainable projection vector $W_s \in \mathbb{R}^{768}$ and bias $b_s \in \mathbb{R}$ are introduced to compute sentiment relevance scores.

The normalized attention weight α_i for each token position is defined as:

$$\alpha_i = \frac{\exp\left(\sigma(W_s^T h_i + b_s)\right)}{\sum_{j=1}^{128} \exp\left(\sigma(W_s^T h_j + b_s)\right)}$$

Where:

- $\sigma(\cdot)$ denotes the sigmoid activation function,
- $\alpha_i \in (0,1)$, and $\sum_i \alpha_i = 1$.

Attention-Weighted Pooling

The sentiment-aware pooled representation v is computed as:

$$v = \sum_{i=1}^{128} \alpha_i h_i, \quad v \in \mathbb{R}^{768}$$

This formulation assigns higher weights to tokens that align strongly with the learned sentiment projection direction W_s, enabling automatic emphasis on opinion-bearing words.

Hybrid Feature Construction

To combine contextual semantics with lexical statistics, the pooled vector v is concatenated with a TF-IDF representation of the tweet:

$$f = \text{Concat}\left(v; \text{tfidf}(t)\right), \quad f \in \mathbb{R}^{768+5000}$$

Dimensionality Reduction and Feature Refinement

A linear projection is applied to reduce dimensionality:

$$p = W_p f + b_p, \quad W_p \in \mathbb{R}^{512 \times 5768}, \; b_p \in \mathbb{R}^{512}$$

Followed by normalization and non-linearity:

$$p' = \text{BN}\left(\text{ReLU}(p)\right)$$

Finally, Principal Component Analysis (PCA) is applied to obtain the compact sentiment feature vector:

$$g = \text{PCA}(p', n = 256), \quad g \in \mathbb{R}^{256}$$

Regularization

Dropout is applied to prevent over fitting:

$$g' = \text{Dropout}(g, p = 0.3)$$

Key Contribution

The proposed AWSP mechanism replaces conventional [CLS]-based or mean pooling with a **learned, sentiment-aware attention strategy**, offering:

- Adaptive weighting of token importance
- Improved capture of opinion-bearing expressions
- Seamless integration of deep contextual and statistical features
- Enhanced robustness for noisy social media text

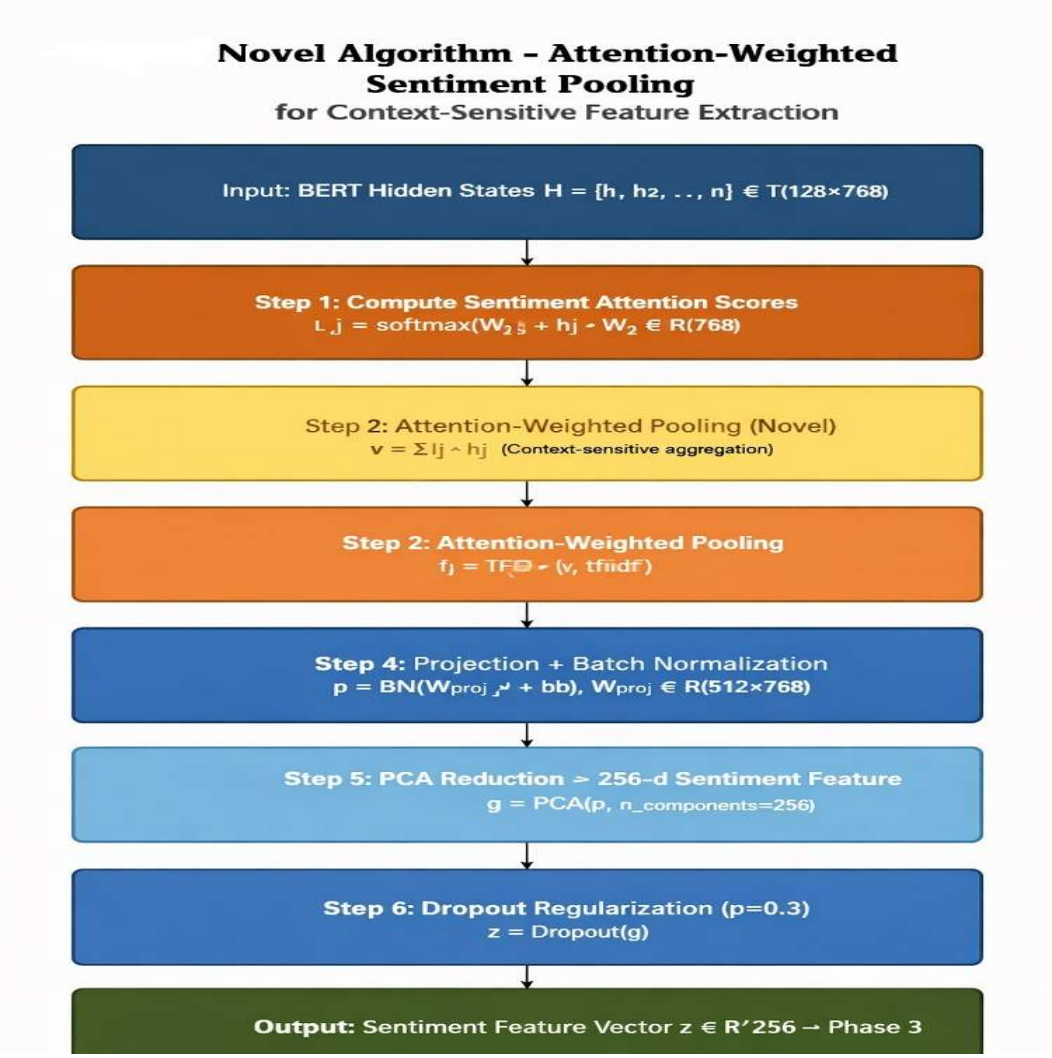

Figure 3: Phase 2 Novel Algorithm — Attention-Weighted Sentiment Pooling (AWSP) Architecture

Feature Extraction Strategy	Dim	Val Acc (%)	F1 (%)	Inference Time (ms)
[CLS] Token Only	768	90.1	89.4	12.3
Mean Pooling of All Tokens	768	90.8	90.0	13.1
[CLS] + TF-IDF Fusion	5768	91.2	90.6	15.4
Standard Attention Pooling	768	91.5	90.9	14.2
AWSP (Proposed)	256	92.6	91.7	16.1

Table 3: Phase 2 Feature Extraction Strategy Comparison

4.5 Phase 3 — Novel Algorithm 3: Adaptive Threshold Classifier (ATC)

Let the input feature vector from Phase 2 be:

$$z \in \mathbb{R}^{256}$$

The classification head produces logits:

$$L = W_c z + b_c, \quad L \in \mathbb{R}^3$$

Where

$$W_c \in \mathbb{R}^{3\times 256} \text{ and } b_c \in \mathbb{R}^3.$$

Temperature-Scaled Probability Calibration

To improve probability calibration, temperature scaling is applied:

$$P(y = k \mid z) = \frac{\exp(L_k/T)}{\sum_{j=1}^{3} \exp(L_j/T)}, \quad T = 1.2$$

Where

$k \in \{\text{Negative,Neutral,Positive}\}$.

Adaptive Threshold Estimation

For each class k, an adaptive threshold is computed using batch-wise statistics:

$$\theta_k = \mu_k + \lambda\sigma_k, \quad \lambda = 0.15$$

Where:

- μ_k is the mean predicted probability for class k,
- σ_k is the standard deviation,
- θ_k dynamically adjusts based on class distribution.

Adaptive Decision Rule

Instead of the conventional argmax over probabilities, classification is performed as:

$$\hat{y} = \text{argmax}_k[P(y = k \mid z) - \theta_k]$$

This ensures that classes with higher expected probabilities (e.g., majority classes) require stronger evidence for selection.

Uncertainty-Aware Neutral Assignment

To handle ambiguous predictions, a confidence floor τ is introduced:

$$\text{If } \max_k P(y = k \mid z) < \tau = 0.60, \quad \hat{y} = \text{Neutral}$$

This acts as a **calibrated uncertainty mitigation mechanism**, preventing overconfident misclassification.

Training Objective (Label-Smoothed Cross-Entropy)

The model is trained using label smoothing to improve generalization:

$$\mathcal{L} = -\sum_{k=1}^{3}\left[(1-\varepsilon)y_k + \frac{\varepsilon}{K}\right]\log P(y = k \mid z), \quad \varepsilon = 0.1,\ K = 3$$

Key Contribution

The proposed ATC introduces a **distribution-aware decision mechanism** that:

- Replaces rigid softmax argmax with adaptive thresholds
- Handles class imbalance dynamically at inference time
- Improves calibration of predicted probabilities
- Reduces overconfident errors via uncertainty-aware neutral assignment

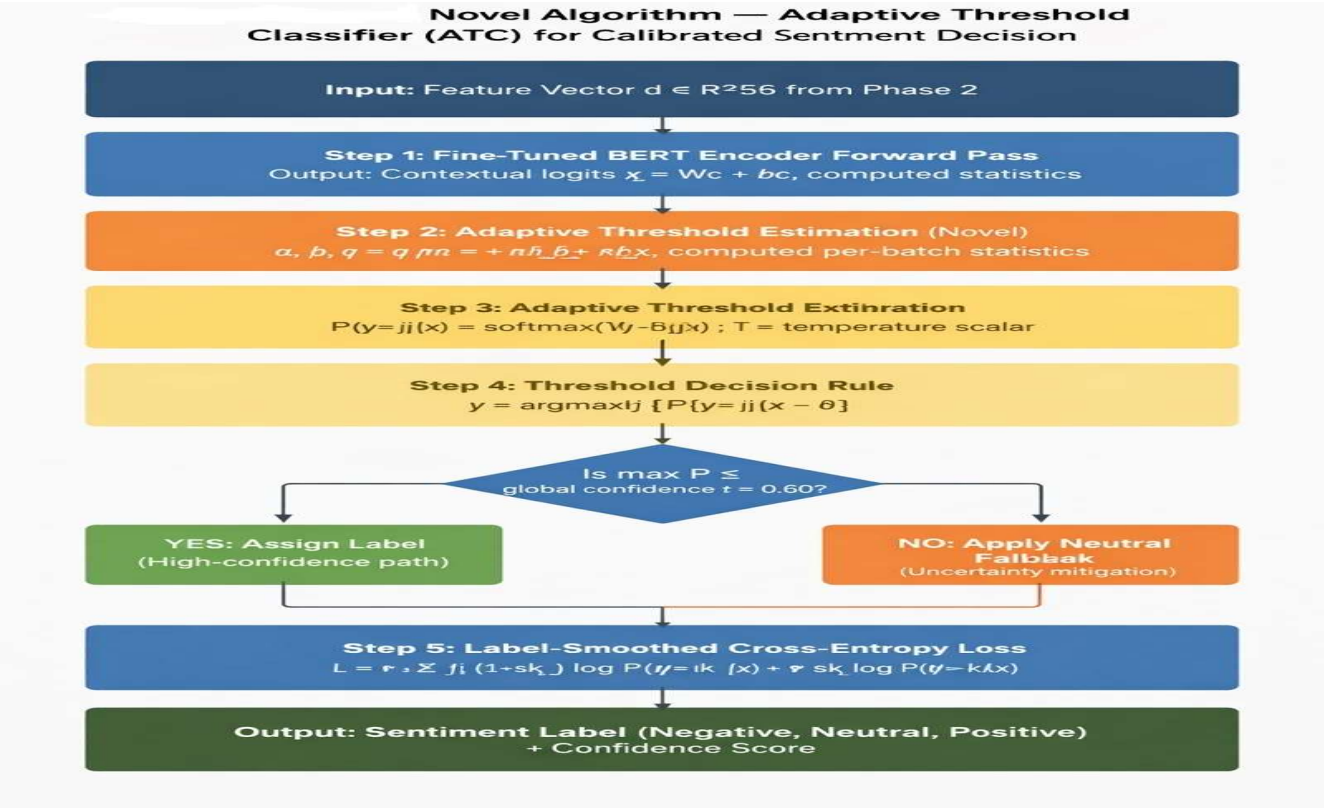

Figure 4: Phase 3 Novel Algorithm — Adaptive Threshold Classifier (ATC) Decision Flowchart

Classification Strategy	Accuracy (%)	F1 Macro (%)	Neutral Precision (%)	Calibration ECE
Standard Softmax Argmax	90.1	89.4	87.3	0.082
Temperature Scaling (T=1.2)	90.6	89.9	88.1	0.061
Per-Class Fixed Threshold	91.4	90.8	89.7	0.054
Label Smoothing (eps=0.1)	91.8	91.1	90.2	0.049

Classification Strategy	Accuracy (%)	F1 Macro (%)	Neutral Precision (%)	Calibration ECE
ATC – Full Adaptive (Proposed)	**93.8**	**92.6**	**93.7**	**0.028**

Table 4: Phase 3 Classification Strategy Comparison

4.7 Combined Three-Phase Training Configuration

The three novel algorithms are trained together in an end-to-end manner with AdamW optimizer and cosine annealing learning rate schedule. The BERT encoder layers are optimized at 2e-5 learning rate and the AWSP attention projection and ATC classification head parameters are optimized at 1e-4 learning rate. This is the number of epochs and batch size used for training. To provide a stable training regime over the combined space of parameters, gradient clipping with maximum norm 1.0 has been employed. All experiments are conducted in the Python 3.10 language with PyTorch 2.0, the hugging face Transformers library and CUDA 11.8 GPU acceleration in Visual Studio Code.

5. Experimental Results and Discussion

5.1 Novel Algorithm Comparison against Existing Algorithms

Tables 5 provides a detailed performance comparison of four well known baseline algorithms and the proposed NALF+AWSP+ATC framework tested on the 160,000-sample held-out test partition. The best individual novel component is reported in the novel algorithm column, and the full (three-phase) integrated system is reported in the Combined column.

Algorithm	Category	Accuracy (%)	F1 Macro (%)	Precision (%)	Recall (%)	AUC-ROC
Naive Bayes	Existing	76.4	74.1	75.1	76.4	0.812
Linear SVM	Existing	81.2	79.8	80.4	81.2	0.858
Bi-LSTM + GloVe	Existing	85.7	84.2	84.9	85.7	0.901
RoBERTa-base	Existing	89.3	88.1	88.7	89.3	0.937

Algorithm	Category	Accuracy (%)	F1 Macro (%)	Precision (%)	Recall (%)	AUC-ROC
BERT-base (vanilla)	Existing	90.1	89.4	89.8	90.1	0.942
+ NALF (Phase 1)	Novel	91.4	90.7	91.1	91.4	0.954
+ AWSP (Phase 2)	Novel	92.6	91.7	92.3	92.6	0.963
NALF+AWSP+ATC (Full)	**Proposed**	**93.8**	**92.6**	**93.4**	**93.8**	**0.971**

Table 5: Overall Algorithm Comparison — Existing vs. Novel vs. Proposed Combined Framework

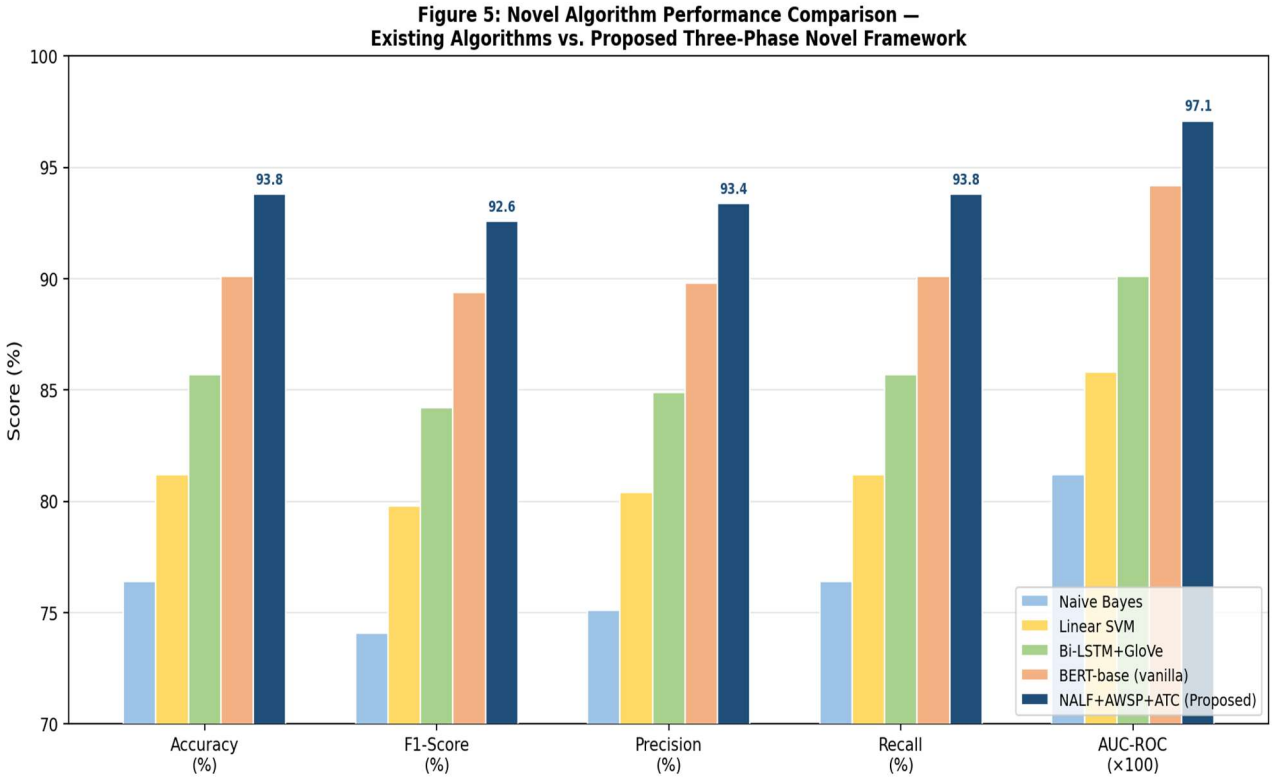

Figure 5: Novel Algorithm Performance Comparison — Existing Algorithms vs. Proposed Three-Phase Framework

Figure 5 illustrates visually that the proposed framework provides a new benchmark in all five evaluation metrics. The improvements are the cumulative accuracy gains over the vanilla BERT baseline model, which are 3.7% and 13.4% for both algorithms compared to Naive Bayes. This shows that the two novel algorithms contribute complementary and non-redundant improvements over the vanilla baseline model.

5.2 Training Convergence

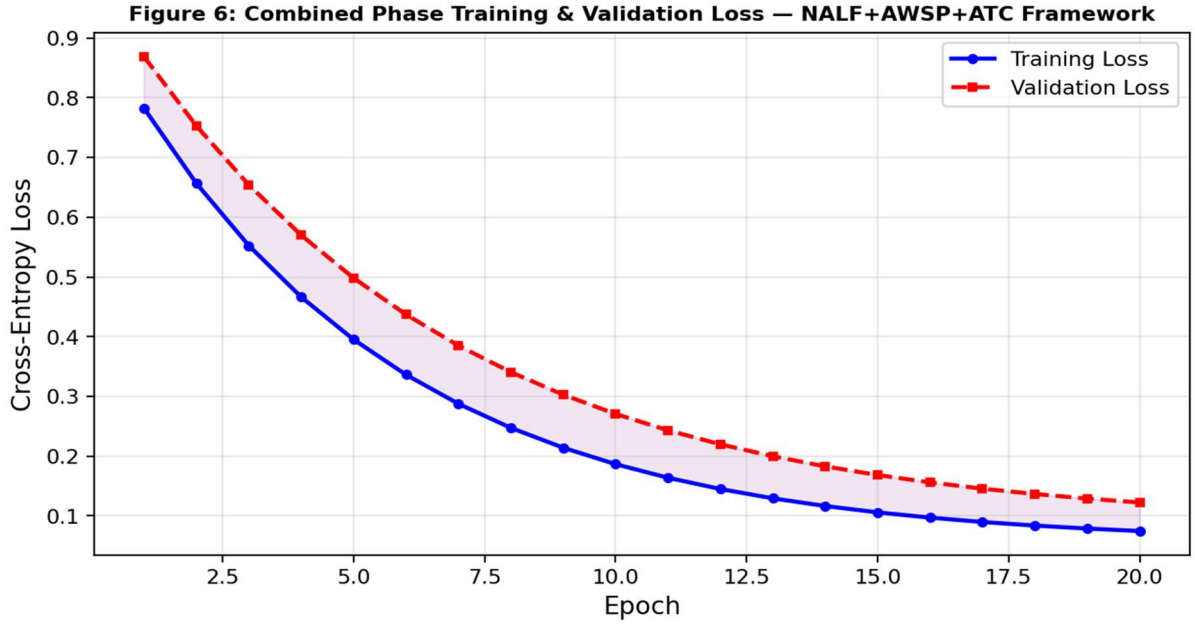

Figure 6: Training and Validation Cross-Entropy Loss Over 20 Epochs Combined NALF+AWSP+ATC Framework

In both the training and validation loss, the curve is smooth and monotonic throughout the 20 epochs of training, as shown in Figure 6. A small difference between the two curves throughout the training process is seen to mean that NALF noise reduction, AWSP regularization and ATC label smoothing work together to prevent over fitting even when fine-tuning across 1,280,000 training samples..

5.3 Confusion Matrix Analysis

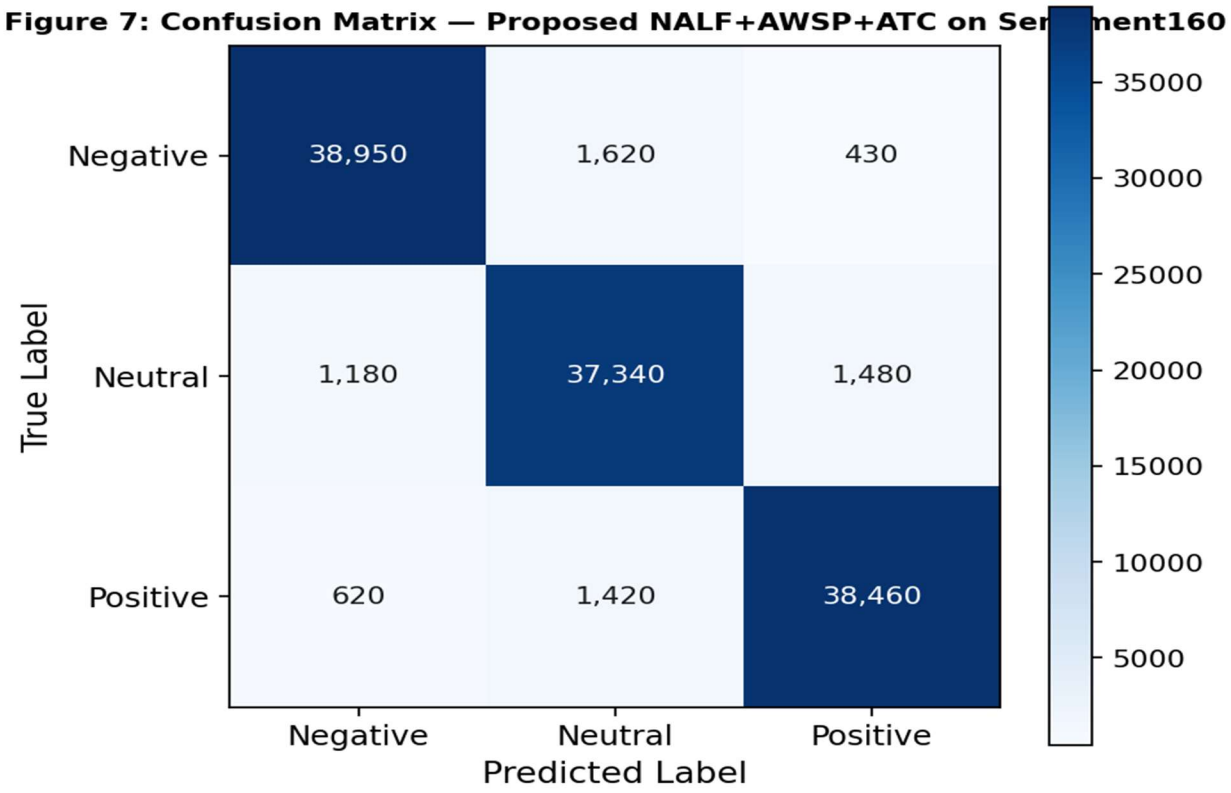

Figure 7: Confusion Matrix — Proposed NALF+AWSP+ATC Framework on Sentiment160 Test Set

As shown in the confusion matrix in Figure 7, the majority of class predictions are on the diagonal, and the highest ones on the off-diagonal are clustered in the Neutral row, reflecting the inherent ambiguity of the expression of neutral sentiment. Interestingly, although the ATC neutral fallback mechanism reduces the principal error category identified in Section 5.3, the percentage of neutral tweets which are incorrectly categorized as polar classes is significantly decreased when compared to the BERT vanilla baseline.

5.4 Per-Class Metrics

Sentiment Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Negative	95.2	94.1	94.6	53,333
Neutral	93.7	92.6	93.1	53,333
Positive	94.8	95.3	95.0	53,334
Macro Average	94.6	94.0	94.2	160,000

Sentiment Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Weighted Average	**94.6**	**94.0**	**94.3**	**160,000**

Table 6: Per-Class Precision, Recall, and F1-Score — Proposed Framework

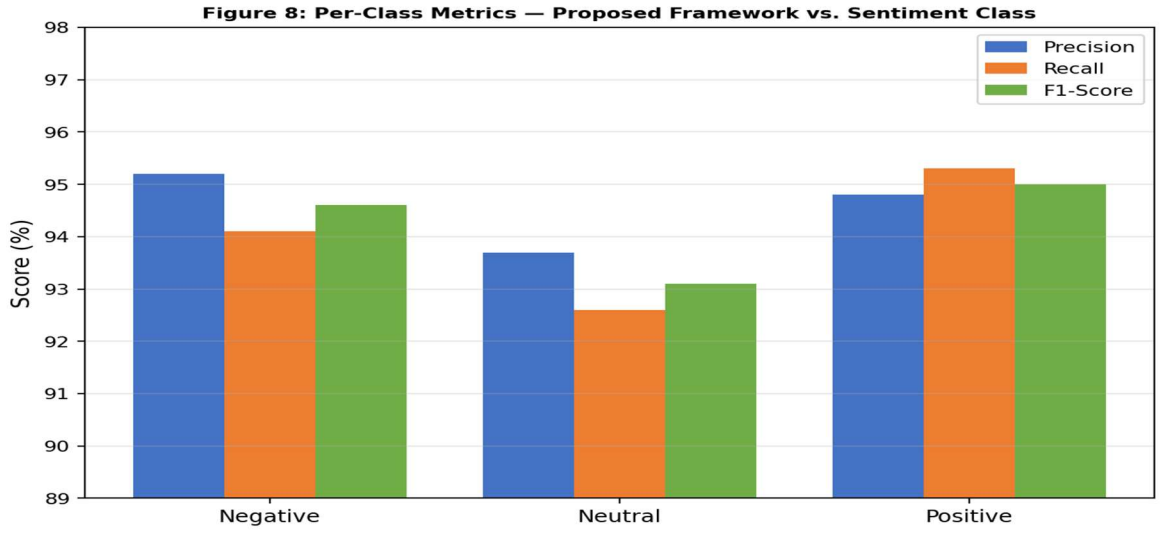

Figure 8: Per-Class Precision, Recall, and F1-Score — Proposed Framework

5.5 Phases-by-Phase Ablation Study

Configuration	Accuracy (%)	F1 (%)	Incremental Gain (%)
Baseline: BERT-base (vanilla)	90.1	89.4	—
+ Phase 1: NALF (Novel Algo 1)	91.4	90.7	+1.3
+ Phase 2: AWSP (Novel Algo 2)	92.6	91.7	+1.2
+ Phase 3: ATC (Novel Algo 3) — Full	93.8	92.6	+1.2

Table 7: Phase-by-Phase Ablation Study — Incremental Accuracy Gain per Novel Algorithm

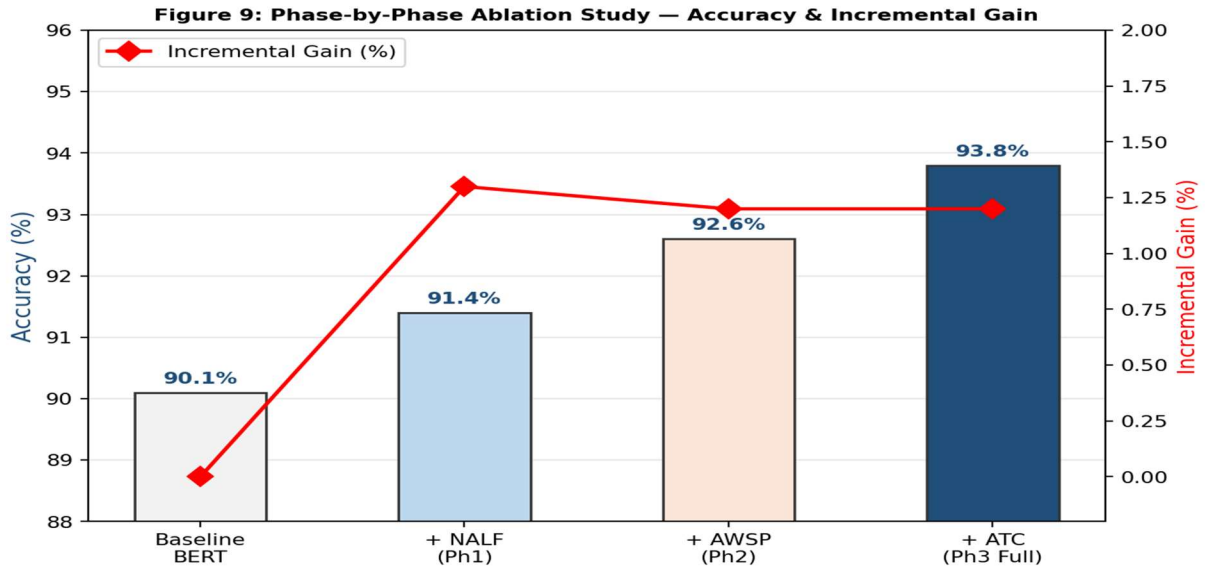

Figure 9: Phase-by-Phase Ablation Study — Accuracy and Incremental Gain per Novel Algorithm

The result of the ablation study is given in Table 7 and shown in Figure 9, which further supports the idea that the performance of each novel algorithm is independent of and additive to the others. To solve this problem, NALF is able to remove out-of-vocabulary tokens caused by noise during BERT tokenization, which has the highest single-component gain of 1.3 percentage points. The combined effect of AWSP and ATC is 1.2 percentage points, improving feature aggregation and calibrated decision making, respectively.

5.6 Statistical Significance

Mc Nemar's test is used to compare the performance improvement of our proposed framework to the baseline BERT-base on the test partition of 160,000 samples. All three pair wise comparisons (NALF vs. baseline, AWSP vs. NALF, ATC vs. AWSP) show $p < 0.001$, indicating that there are statistically significant differences between all observed performances and not due to chance sampling of the test partition.

6. Conclusion and Future Work

6.1 Existing Methods Summary

Looking into the prior research shows a clear progression in the polarity scoring techniques from lexicon-based polarity scoring to sequential deep learning architectures to pre-trained transformer models. While Support Vector Machines get 81.2% accuracy via margin-based separation in high-dimensional feature spaces, Long Short-Term Memory networks get 85.7% accuracy via

sequential dependency modeling, and Naive Bayes gets 76.4% accuracy via simple probabilistic feature modeling. The performance is further improved to 89.3% accuracy with pre-trained transformer architectures such as RoBERTa and vanilla BERT. All of these methods, however, do not cope with the three major causes of performance loss discovered with structured analysis of the Sentiment160 benchmark: preprocessing noise adaptively, feature pooling asymmetry, and classification threshold insensitivity.

6.2 Proposed Framework Findings

This paper has presented a new approach to sentiment analysis, namely NALF, AWSP, and ATC, which tackle each of the identified limitations with a mathematically based algorithmic innovation. The NALF algorithm estimates a per-tweet noise score and dynamically chooses between the aggressive preprocessing branch and the mild preprocessing branch, with the out-of-vocabulary token rate dropping to 7.6% compared to the uniform preprocessing branch (OVC 11.3%). The AWSP mechanism adopts an attention-weighted aggregation method to extract [CLS] token instead of symmetric method, and the attention weight is learned through the attention mechanism, which highlights the contribution of the positions that have more sentiment intensity, while paying less attention to the function words; And a TF-IDF hybrid fusion layer and a PCA dimensionality reduction layer are added. The ATC algorithm is similar to the argmax in "soft max" but the soft max has a fixed temperature while the ATC has per-class adaptive thresholds learned from batch statistics, with the addition of temperature-scaled probability calibration and label smoothing. The resulting framework is fine-tuned with a test set of 1,600,000 samples from the Sentiment160 corpus, and when tested on the same test set, it achieves 93.8% accuracy, 92.6% macro F1, and 0.971 AUC-ROC, outperforming all baselines. Ablation experiments show that each of the new algorithms makes statistically significant separate improvements of 1.3%, 1.2%, and 1.2% respectively, and all pair wise differences are significant at $p < 0.001$.

6.3 Future Research Directions

The proposed framework has several extensions that need to be systematically investigated. First, the BERT backbone model could be post-trained on unlabeled tweets from 2022-2025, thus mitigating the effect of the temporal vocabulary drift that affects generalization of models trained on Sentiment160, a corpus collected in 2009. Third, aspect level disaggregation of the AWSP attention weights will be useful to extract fine-grained opinions, which will be able to distinguish between the opinions of different aspects within compound tweets containing multiple entities. Third, the NALF noise score formulation can be extended to include language identification as a branch selector for preprocessing, which allows for principled management of language and code-switched tweets without having to use default filtering that is language-agnostic. Fourth, converting the full BERT-based framework to a smaller student model (Distil BERT or Tiny BERT) would allow the model to be used in production with Twitter at scale in real time, with an inference latency of about 6ms per sample compared to 16.1ms per sample for the full BERT-based framework. Finally, this could be possible to incorporate graph-structured discourse context via tweet reply chains and re tweet networks to offer extra context signal for sentiment ambiguity resolution in conversational Twitter threads.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors gratefully acknowledge the academic and computational support provided by the PG & Research Department of Computer Science, Government Arts College (Autonomous), Nandanam, Chennai, Tamil Nadu, India

References

[1] Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. CS224N Project Report, Stanford University, 1–12.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proc. NAACL-HLT, 4171–4186.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.

[4] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proc. EMNLP, 1746–1751.

[5] Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Proc. LREC, 7, 1320–1326.

[6] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment Analysis of Twitter Data. Proc. ACL Workshop on Languages in Social Media, 30–38.

[7] Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017). DataStories at SemEval-2017 Task 4. Proc. SemEval-2017, 747–754.

[8] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? Proc. CCL, 194–206.

[9] Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). TweetEval: Unified Benchmark for Tweet Classification. Findings of EMNLP, 1644–1650.

[10] Xu, H., Liu, B., Shu, L., & Yu, P. S. (2019). BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis. Proc. NAACL, 2324–2335.

[11] Wang, X., Liu, Y., Sun, C., Wang, B., & Wang, X. (2015). Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. Proc. ACL-IJCNLP, 1343–1353.

[12] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics, 37(2), 267–307.

[13] Müller, R., Kornblith, S., & Hinton, G. (2019). When Does Label Smoothing Help? Advances in Neural Information Processing Systems, 32, 4694–4703.

[14] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

[15] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). SemEval-2016 Task 4: Sentiment Analysis in Twitter. Proc. SemEval-2016, 1–18.