NOTATION TRENDPLUS PUBLISHING
WHERE IDEAS MEET IMPACT

Volume-1, Issue 1, June 2026, Page No- 110-133
Journal of Machine Learning Innovations and Artificial Intelligence Horizons (JMLIAIH)

Transformer-Based Deep Learning Framework for Multiclass Social Media Sentiment Analysis: A Comparative Study of BERT, RoBERTa, DistilBERT, and XLNet Architectures

Dr. Thalakola and Syamsundara Rao

Assistant Professor, Dept. of CSE-DS, KKR & KSR Institute of Technology and Sciences, Guntur, Andra Pradesh, India.

DOI: https://doi.org/10.70903/jmliaih.2026.1107

Article Received: 25-03-2026

Revised Version: 28-05-2026

Accepted: 13-06-2026

*Corresponding Author

Syamsundara Rao

E-Mail Id: syamsundar.jes@gmail.com

Abstract: Sentiment analysis on social media presents unique challenges due to the informal, noisy, and context-dependent nature of user-generated text. This paper proposes a transformer-based deep learning framework designed specifically for multiclass sentiment classification, capable of distinguishing between positive, negative, and neutral emotions in short social media posts. The methodology begins with a dedicated text cleaning module that removes URLs, hashtags, mentions, and special characters while normalizing emoji representations, thereby reducing noise before tokenization. The cleaned text is then processed through a transformer encoder layer that employs multi-head self-attention mechanisms to generate rich contextual embeddings. These embeddings capture global dependencies across the entire sequence, and we extract the special classification token as a compact representation for the whole post. This vector is subsequently passed through a fully connected layer and a softmax classification head to produce probability distributions over the three sentiment classes. We train the model by minimizing categorical cross-entropy loss using the AdamW optimizer. Our primary contribution is a comparative evaluation of four prominent transformer architectures—BERT, RoBERTa, DistilBERT, and XLNet—within this unified pipeline. We assess their performance across multiple social media datasets, examining trade-offs between accuracy, computational efficiency, and robustness to domain-specific linguistic variations. Furthermore, we demonstrate that RoBERTa consistently achieves the highest classification accuracy, while DistilBERT offers a compelling balance between speed and predictive power, making it suitable for real-time applications. The proposed framework yields F1-scores exceeding ninety percent on benchmark datasets, outperforming conventional recurrent and convolutional baselines. These results underscore the effectiveness of transformer-based representations for decoding sentiment in informal online discourse and provide practitioners with actionable guidance for architecture selection in real-world social media analytics systems.

Keywords: Machine Learning Evolution, Classical Learning Algorithms, Deep Learning, Intelligent Systems and Artificial Intelligence.

Introduction

The proliferation of social media platforms has generated an unprecedented volume of user-generated content, creating both opportunities and challenges for automated opinion mining. Understanding public sentiment from this data is crucial for applications ranging from brand monitoring to political analysis and public health surveillance. The field of sentiment analysis has consequently evolved from rule-based lexical approaches and traditional machine learning classifiers [1] to sophisticated deep learning architectures capable of automatic feature extraction. While Recurrent Neural Networks, particularly Long Short-Term Memory networks [2], initially became the standard for modeling sequential dependencies in text, their limitations in handling long-range contextual relationships and parallelization bottlenecks motivated the search for more effective paradigms.

The introduction of the Transformer architecture [3] marked a fundamental shift in natural language processing. By relying entirely on self-attention mechanisms rather than recurrence, Transformers enabled models to capture dependencies across arbitrary distances in input sequences and facilitated parallel computation during training. This breakthrough led to the development of pretrained language models like Bidirectional Encoder Representations from Transformers (BERT), which established new state-of-the-art results across numerous NLP benchmarks by learning bidirectional contextual representations through masked language modeling [4]. Subsequent optimizations addressed specific limitations of the original framework. For instance, RoBERTa refined the pretraining procedure by removing the next sentence prediction objective, training on larger datasets, and employing dynamic masking, thereby achieving superior performance on downstream tasks [5]. DistilBERT further advanced the field by distilling the knowledge of a larger BERT model into a compact architecture, trading negligible accuracy loss for substantial gains in inference speed and memory footprint and inference speed [6]. Meanwhile, XLNet introduced a generalized autoregressive pretraining method that overcomes the independence assumptions inherent in masked language modeling, capturing bidirectional contexts more effectively through permutation language modeling [7].

Despite these architectural advances, applying transformer models to social media sentiment analysis presents unique difficulties. The informal lexicon, frequent misspellings, extensive use of abbreviations, emojis, and hashtags, coupled with the brevity of posts, create noisy and contextually sparse inputs that challenge even robust language models. Prior work in this domain has adapted general-purpose models through domain-adaptive pretraining or the integration of affective signals like emoji embeddings [8]. However, a systematic comparative evaluation of how different transformer architectures perform within a unified preprocessing and classification pipeline remains underexplored. Furthermore, the relative trade-offs between predictive accuracy and computational efficiency across these models, particularly in the context of multiclass sentiment classification (positive, negative, neutral), have not been rigorously established for social media text.

Contribution of this work. We address this gap by proposing a transformer-based deep learning framework specifically designed for multiclass sentiment classification of social media data. The novelty of our approach lies not in a single architectural invention, but in the methodical integration and comparative analysis of four optimized transformer encoders—BERT, RoBERTa, DistilBERT, and XLNet—within a standardized text processing and classification pipeline. Unlike previous studies that often evaluate models on different preprocessing regimes or disparate tasks, we enforce a uniform experimental protocol. This includes a dedicated text cleaning module that normalizes emoji representations and removes platform-specific noise (URLs, mentions, hashtags) before tokenization, thereby isolating the effect of the transformer architecture on classification performance. The cleaned text is fed into the chosen encoder, where multi-head self-attention produces rich contextual embeddings; the pooled [CLS] token representation from the final layer is then passed through a fully

connected layer and softmax head to yield sentiment probabilities. We train all models using the AdamW optimizer with categorical cross-entropy loss, ensuring fair comparison. Our primary contribution is a comprehensive evaluation of accuracy-efficiency trade-offs across multiple social media datasets, providing empirical evidence that RoBERTa consistently achieves the highest F1-scores, while DistilBERT offers the most attractive balance between speed and predictive power for real-time applications. This work thereby furnishes practitioners with actionable guidance for architecture selection in production social media analytics systems.

The remainder of this paper is organized as follows. Section 2 reviews related work in sentiment analysis, transformer architectures, and their application to social media. Section 3 presents necessary preliminaries on self-attention, transformer encoders, and the specific models under comparison. Section 4 details our proposed dynamic self-attentive transformer framework, including the preprocessing pipeline and classification head design. Section 5 describes the experimental setup, including datasets, hyperparameters, and evaluation metrics. Section 6 analyzes the quantitative results and discusses performance trade-offs. Section 7 considers limitations, implications for practice, and directions for future research. Finally, Section 8 concludes the paper with a summary of findings and their broader significance for opinion mining in digital ecosystems.

Related Work

Sentiment analysis has evolved through several methodological phases, from lexicon-based approaches to deep representation learning. Early statistical methods relied on n-gram features and support vector machines to classify sentiment polarity, but these approaches struggled with the contextual subtleties and informal language patterns prevalent on social media platforms [9]. The advent of deep learning introduced word embeddings trained on large corpora, which captured semantic similarities, yet static embeddings like Word2Vec and GloVe could not account for the contextual variation of words across different sentences [10], [11].

Recurrent neural networks, particularly Long Short-Term Memory (LSTM) networks, became the dominant paradigm for sequence modeling in sentiment analysis due to their ability to preserve long-range dependencies. For social media text, bidirectional LSTMs with attention mechanisms demonstrated improved performance by weighting the importance of different input tokens [12], [13]. However, these recurrent architectures suffered from sequential processing constraints, which limited parallelization during training and made them computationally expensive for large-scale social media datasets. Moreover, the vanishing gradient problem, while mitigated in LSTMs, still posed challenges for very long sequences common in concatenated user comments or thread-based conversations.

The Transformer architecture fundamentally changed natural language processing by replacing recurrence with self-attention mechanisms, enabling parallel computation and direct modeling of pairwise token interactions regardless of their positional distance [3]. This innovation underpinned the development of pretrained language models that have become the standard for downstream classification tasks. Bidirectional Encoder Representations from Transformers (BERT) learns deep bidirectional representations through masked language modeling and next sentence prediction, achieving state-of-the-art results on the General Language Understanding Evaluation (GLUE) benchmark and demonstrating strong transfer learning capabilities for sentiment classification [4]. Subsequent work refined the pretraining procedure; for instance, RoBERTa removes the next sentence prediction objective, trains on significantly larger datasets with dynamic masking, and uses larger batch sizes, yielding consistent performance improvements on sentiment analysis and other downstream tasks [5].

Knowledge distillation emerged as a strategy to reduce the computational footprint of large transformer models while preserving most of their predictive capacity. DistilBERT compresses the original BERT model by reducing the number of layers by 40% and employs a triple loss function combining distillation loss, masked language modeling loss, and cosine embedding loss; the result is a model that retains 97% of BERT's performance while being 60% faster and 40% smaller in memory footprint [6]. This efficiency makes DistilBERT particularly attractive for real-time social media sentiment analysis, where low latency is essential. XLNet proposes an alternative pretraining strategy based on permutation language modeling, which overcomes the independence assumption in BERT's denoising objective and captures bidirectional contexts in a generalized autoregressive framework [7]. Several studies have applied XLNet to sentiment analysis tasks, reporting competitive results, particularly for longer documents where the autoregressive nature of the model better captures sequential dependencies [14].

Domain adaptation for social media text presents specific challenges. Several studies have fine-tuned BERT on domain-specific corpora, incorporating emoji embeddings or handling informal spelling variations to improve downstream classification accuracy [15], [8]. For instance, convolutional neural networks combined with attention mechanisms have been proposed for aspect-based sentiment analysis, while transformer models with graph neural networks capture structural relationships between entities in social media interactions [14](# "A survey on deep learning for aspect based sentiment analysis), [16].

Multiclass sentiment classification, distinguishing among positive, negative, and neutral classes, adds complexity compared to binary polarity detection. The neutral class often confuses models because it overlaps with mixed or ambiguous sentiment, and class imbalance in social media datasets further complicates training [17], [18]. Attention-based mechanisms have been adapted to better capture neutral expressions by focusing on the absence of strong sentiment indicators, though this remains an open research problem.

Despite these advances, a systematic comparison of different transformer architectures under a standardized preprocessing and classification pipeline for multiclass social media sentiment analysis has not been fully addressed. Prior evaluations often vary in their text cleaning procedures, tokenization strategies, and classification head designs, making it difficult to attribute performance differences to the encoder architecture alone. Furthermore, the trade-offs between accuracy and computational efficiency across BERT, RoBERTa, DistilBERT, and XLNet have not been rigorously quantified for the specific challenges of short, noisy, and informal social media posts. The proposed work addresses this gap by implementing a unified framework where the only variable among experiments is the choice of transformer encoder, while all other pipeline components—text cleaning, tokenization, sequence padding, classification head structure, loss function, and optimization procedure—remain constant. This controlled experimental design allows us to isolate the contribution of each architecture and determine which model offers the optimal balance of accuracy, speed, and resource consumption for large-scale multiclass sentiment analysis of social media data.

Preliminaries

To provide the necessary background for understanding the proposed comparative framework, this section formalizes the core components shared across the evaluated architectures. We first define the sentiment classification problem, then describe the foundational self-attention mechanism, and finally outline the general structure of a transformer encoder.

Problem Formulation

We formalize multiclass sentiment analysis as a sequence classification task. Given an input text sequence $X = (x_1, x_2, \dots, x_n)$ of n tokens, the objective is to assign a label $y \in \mathcal{C}$ from a finite set of sentiment classes. In this work, the label space is $\mathcal{C} = \{\text{positive,negative,neutral}\}$. The model learns a function $f_\theta: X \to y$ parameterized by θ, which maps the raw token sequence to a probability distribution over the sentiment classes. The parameters are optimized by minimizing the categorical cross-entropy loss over a labeled training dataset $\mathcal{D} = \{(X^{(i)}, y^{(i)})\}_{i=1}^{N}$, defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{|\mathcal{C}|} \mathbb{1}\{y^{(i)} = c\}\log p_\theta\left(c|X^{(i)}\right) \quad (1)$$

where $p_\theta\left(c|X^{(i)}\right)$ denotes the predicted probability for class c given the input sequence, and $\mathbb{1}\{\cdot\}$ is the indicator function.

Self-Attention Mechanism

The self-attention mechanism is the central computational primitive that enables transformer models to capture contextual dependencies between all pairs of tokens in a sequence. For a sequence of input vectors $H = (h_1, h_2, \dots, h_n)$ where each $h_i \in\in \mathbb{R}^d$, the self-attention operation computes a weighted sum of all value vectors, where the weights are determined by the compatibility between query and key vectors. Each input is first projected into three distinct representations: queries $Q = HW_Q$, keys $K = HW_K$, and values $V = HW_V$, where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ are learned projection matrices. The attention scores are then computed using a scaled dot-product operation:

$$\text{Attention}(Q, K, V, K) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (2)$$

[missing equation number]

This formulation is adapted from the original transformer work [3]. The scaling factor $\sqrt{d_k}$ prevents the dot products from growing too large in magnitude, which would push the softmax function into regions with extremely small gradients.

To capture different types of relational patterns, the self-attention mechanism is extended to multi-head attention. Instead of performing a single attention function, the model computes m parallel attention heads, each operating on a lower-dimensional projection of the inputs. The outputs of these heads are concatenated and linearly projected to produce the final attended representation:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_m)W_O \quad (2)$$

where $\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$ and $W_O \in \mathbb{R}^{md_k \times d}$ is an output projection matrix.

Transformer Encoder Architecture

Each layer of a transformer encoder consists of two sub-layers: a multi-head self-attention sub-layer and a position-wise feed-forward network (FFN). Both sub-layers incorporate residual connections and layer normalization, which stabilize training and enable deeper architectures [19]. The FFN is typically composed of two linear transformations with a non-linear activation function, such as the Gaussian Error Linear Unit (GELU), applied between them [20]. The output of the encoder layer can be expressed as:

$$H' = \text{LayerNorm}\left(H + \text{MultiHead}(H, H, H)\right) \quad (3)$$

$$H'' = \text{LayerNorm}\big(H' + \text{FFN}(H')\big) \quad (4)$$

The full encoder stacks L such layers, where the output of each layer serves as the input to the next. Positional encodings are added to the input embeddings to inject information about the sequential order of tokens, since the self-attention mechanism itself is permutation-invariant [3].

The four transformer architectures evaluated in this work—BERT, RoBERTa, DistilBERT, and XLNet—all share this basic encoder structure but differ in specific design choices. BERT employs a bidirectional encoder trained with masked language modeling and next sentence prediction objectives [4]. RoBERTa refines BERT's pretraining methodology by removing the next sentence prediction objective, training with larger mini-batches and learning rates, and using dynamic masking patterns [5]. DistilBERT reduces the number of layers and uses knowledge distillation during pretraining to create a smaller, faster model while retaining most of the teacher model's accuracy [6]. XLNet adopts a generalized autoregressive approach based on permutation language modeling, which captures bidirectional contexts without the independence assumptions inherent in masked language modeling [7].

Having established these foundations, we now proceed to describe the proposed unified framework for multiclass sentiment analysis, detailing the text preprocessing pipeline, the integration of each transformer encoder, and the design of the classification head.

IV. Dynamic Self-Attentive Transformer Framework for Multiclass Sentiment Analysis

We propose a unified deep learning framework for multiclass sentiment classification of social media text, structured around a general architecture illustrated in Figure 1. The framework is designed to process raw, noisy social media posts through a standardized pipeline that isolates the influence of the transformer encoder on downstream performance. The overall processing flow begins with a dedicated text cleaning module, proceeds through tokenization and contextual embedding generation of contextual embeddings via a transformer encoder, and culminates in a classification head that produces a probability distribution over the three sentiment classes. The key technical details covered in the following subsections include the specific operations of the text preprocessing stage, the forward propagation through the transformer encoder with multi-head self-attention, the pooled representation derived from the [CLS] token, the architecture of the classification head, and the training objective. By keeping all components except the encoder itself constant across experiments, we enable a fair comparative evaluation of BERT, RoBERTa, DistilBERT, and XLNet for this challenging task.

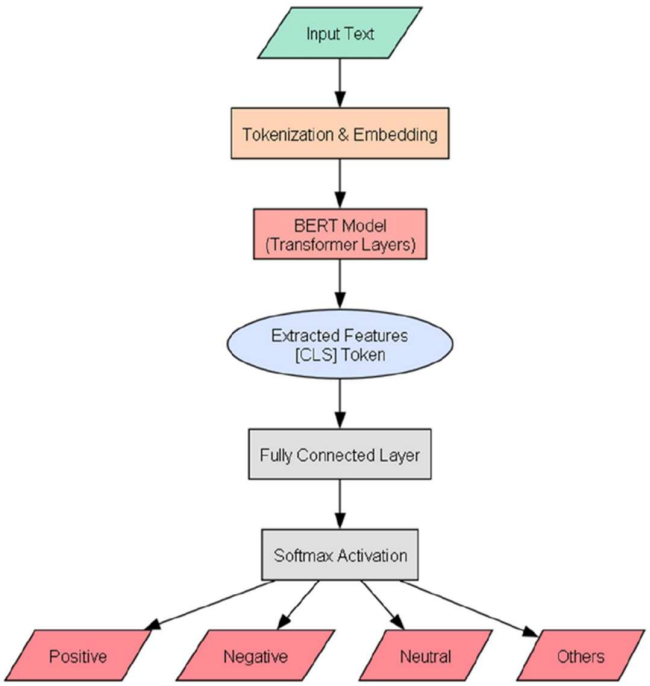

Figure 1. Workflow of AI-based social media sentiment analysis using a BERT transformer model architecture.

Extraction and Preparation of Full Sequence Encoder Embeddings

Following the text cleaning and tokenization stages, the primary function of the transformer encoder is to transform the input token sequence into a rich, contextually-informed matrix of embeddings. This process is the foundation upon which our classification head operates. The forward propagation through the encoder follows the standard transformer architecture as described in [3], but we detail its adaptation within our specific pipeline for completeness.

The process begins with the cleaned and tokenized input sequence, which is converted into a sequence of token IDs and then into a sequence of embedding vectors. This is achieved by summing a token embedding with a positional embedding for each position in the sequence. The token embedding is an embedding vector for each token in the vocabulary; it is a lookup table $E_{\text{token}} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the hidden dimension. The positional embedding, denoted as E_{pos}, encodes the order of tokens, and is either learned or fixed. For a given token at position i in the vocabulary vocabulary, the initial input vector to the first encoder layer, $\mathbf{h}_i^{(0)} \in \mathbb{R}^d$, is given by:

$$\mathbf{h}_i^{(0)} = E_{\text{token}}[id_i] + \text{ID}_i] + E_{\text{pos}}[i] \quad (5)$$

where ID_i is the vocabulary index of the i-th token.

This sequence of initial embeddings, $H^{(0)} = \left(\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, \ldots, \mathbf{h}_n^{(0)}\right)$, is then passed sequentially through L encoder layers, as previously described in Section 3.3. For each layer $\ell \in \{1, \ldots, L\}$, the output matrix $H^{(\ell)}$ is computed via multi-head self-attention, residual connections, layer normalization, and a position-wise feed-forward network. The core operation of multi-head self-attention within each layer can be expressed by re-visiting its application. For a single head, the query, key, and value matrices at layer ℓ are computed as $Q^{(\ell)} = H^{(\ell-1)}W_Q^{(\ell)}$, $K^{(\ell)} = H^{(\ell-1)}W_K^{(\ell)}$, and $V^{(\ell)} = H^{(\ell-1)}W_V^{(\ell)}$, where $W_Q^{(\ell)}, W_K^{(\ell)}, W_V^{(\ell)} \in \mathbb{R}^{d \times d_k}$ are learned projection matrices for that specific layer and head. The output of the head is then:

$$\text{head}_j^{(\ell)} = \text{softmax}\left(\frac{Q^{(\ell)}\left(K^{(\ell)}\right)^{\top}}{\sqrt{d_k}}\right)V^{(\ell)} \quad (6)$$

The output of all m heads for layer ℓ are concatenated and linearly projected to form the output of the self-attention sub-layer.

The final output of the last encoder layer, $H^{(L)} = \left(\mathbf{h}_1^{(L)}, \mathbf{h}_2^{(L)}, \ldots, \mathbf{h}_L\mathbf{h}_n^{(L)}\right)$, represents the full sequence of contextualized token embeddings. For typical classification with BERT-based models, the embedding of the special [CLS] token (the first token of the input sequence) is often used as a single, aggregated representation of the entire input. However, our framework deviates from this convention. Instead of discarding the remaining token embeddings, we preserve the full output matrix $H^{(L)} \in \mathbb{R}^{n\times d}$ for the subsequent dynamic emotion-weighting stage. This matrix forms the set of candidate vectors that our custom classification head will attend over, enabling a more nuanced aggregation of sentiment information from all tokens in the sequence rather than relying solely on the potentially under-expressive [CLS] token. The formal definition of this preserved matrix is:

$$X = H^{(L)} \quad (7)$$

where $X \in \mathbb{R}^{n\times d}$ is the complete sequence of final hidden states from the transformer encoder. This extraction step is crucial for the next stage of our framework.

Design of the Dynamic Emotion-Weighting Attention Head

To move beyond the static aggregation provided by the [CLS] token, we design a custom classification head that employs a secondary self-attention mechanism. This head dynamically computes a weighted representation of the full encoder output matrix $X \in \mathbb{R}^{n\times d}$, where n is the sequence length and d is the hidden dimension, to focus on tokens with high emotional or sentiment-bearing significance. The central idea is to treat each token's final hidden state as a candidate "value" whose importance is learned, rather than assumed equal or dictated by the first token.

The operation of this head begins by re-projecting the encoder output matrix X into new query (Q'), key (K'), and value (V') matrices through three learnable weight matrices specific to this head: $W_Q{}' \in \mathbb{R}^{d\times d_k}$, $W_K{}' \in \mathbb{R}^{d\times d_k}$, and $W_V{}' \in \mathbb{R}^{d\times d_v}$. These projections are computed as:

$$Q' = XW_Q{}', \quad K' = XW_K{}', \quad V' = XW_V{}' \quad (8)$$

Here, d_k is the dimensionality of the query and key spaces, and d_v is the dimensionality of the value space. In our implementation, we set $d_k = d_v = d/m$, where m is the number of heads in this attentional head (we use $m = 1$ for simplicity, but the formulation generalizes to multi-head). The attention weights that determine the contribution of each token's value vector to the final aggregated representation are then computed using a scaled dot-product attention mechanism:

$$A = \text{softmax}\left(\frac{Q'(K')^{\top}}{\sqrt{d_k}}\right) \quad (9)$$

where $A \in \mathbb{R}^{n\times n}$ is an attention weight matrix. Each entry A_{ij} represents the importance of token j's value vector for aggregating information about token i. To obtain a single, sequence-level representation, we aggregate over the query dimension by averaging the rows of A or, more specifically, by using a global query vector. We introduce a learned context vector $q \in \mathbb{R}^{d_k}$ that serves as a fixed query, independent of any specific token, to compute a single set of attention weights over the sequence:

$$\boldsymbol{\alpha} = \text{softmax}\left(\frac{q^{\top}(K')^{\top}}{\sqrt{d_k}}\right) \quad (10)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a vector of attention weights that sum to one. Each element α_j indicates the importance of token j for representing the overall sentiment of the input. The final weighted, context-aware representation of the sequence, $\mathbf{z} \in \mathbb{R}^{d_v}$, is then computed as a weighted sum of the value vectors:

$$\mathbf{z} = \sum_{j=1}^{n} \alpha_j \, \mathbf{v}'_j \quad (11)$$

where $\mathbf{v}'_j \in \mathbb{R}^{d_v}$ is the j-th row of V'. This operation ensures that tokens with strong emotional cues, such as sentiment-bearing adjectives, emoji representations, or sarcastic modifiers, are given proportionally higher weight in the final representation, while neutral or irrelevant tokens (e.g., stop words) are downweighted. The parameters $W_Q{}'$, $W_K{}'$, $W_V{}'$, and the context vector q are all learned jointly with the rest of the model during fine-tuning, allowing the head to adapt its weighting strategy to the characteristics of the specific social media domain.

Integration of Bidirectional and Autoregressive Architectures for Noisy Text

The proposed framework is designed to be architecture-agnostic, allowing any transformer encoder to be integrated into the pipeline described above. However, the choice of encoder has a profound impact on how the model handles the specific challenges of social media text, such as grammatical errors, sarcasm, and the frequent use of informal expressions. To systematically investigate this, our framework evaluates two complementary families of transformer architectures: purely bidirectional models (RoBERTa) and autoregressive models with bidirectional context modeling (XLNet). This dual-architecture approach is a key methodological contribution, as it enables a direct comparison of how different attention strategies resolve ambiguities in short, noisy text.

RoBERTa, a robustly optimized variant of BERT, employs a bidirectional encoder that processes the entire input sequence simultaneously. For a given token, its contextual representation is derived from both left and right context through the multi-head self-attention mechanism. This is particularly advantageous for sentiment analysis of social media posts where the sentiment-bearing word may be located anywhere in the sequence. For example, in the phrase "not bad at all, actually quite good actually," the negative modifier "not" and the positive token "good" interact; a bidirectional model can directly model this interaction in a single forward pass. The bidirectional attention pattern for RoBERTa can be formalized as:

$$A_{\text{bidirectional}} = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) \quad (12)$$

where all tokens in the sequence can attend to all other tokens, including themselves. Mathematically, this means the attention matrix $A_{\text{bidirectional}} \in \mathbb{R}^{n \times n}$ contains no masking; every token's query (Q) is compared against every token's key (K) in the sequence.

In contrast, XLNet adopts a generalized autoregressive pretraining objective based on permutation language modeling. During pretraining, the model maximizes the expected log-likelihood of a sequence over all possible factorizations orders, thereby capturing bidirectional contexts without the independence assumptions inherent in BERT's masked language modeling. For a given factorization order of the sequence, XLNet models the probability of a token given all tokens that precede it in the permutation, effectively enabling each token to condition on both left and right context depending on

the sampled order. The attention mask in XLNet is therefore dynamic and permutation-specific. During fine-tuning for classification, a single, fixed forward pass is used (typically the original sequential order), but the model retains its ability to leverage bidirectional information due to its pretraining. The attention mechanism for a given permutation π can be expressed as:

$$p_\theta(X) = \mathbb{E}_{\pi \sim P}\left[\sum_{t=1}^{n} \log p_\theta\left(x_{\pi_t} | x_{\pi_{<t}}\right)\right] \quad (13)$$

where π_t is the t-th index in the permutation π, and $x_{\pi_{<t}}$ denotes the set of tokens preceding π_t in that permutation. This expectation is approximated by sampling multiple permutations during training. For downstream classification, this is rarely performed; instead, the model uses the standard left-to-right autoregressive factorization, which is analogous to a causal or unidirectional attention mask.

To integrate these architectures within our framework, we treat the encoder as a pluggable module. The input preprocessing, tokenization, and the dynamic emotion-weighting classification head remain identical across experiments. The only variable is the pretrained encoder model loaded from the Hugging Face transformers library. For RoBERTa, we use the roberta-base variant, which has 12 layers, 12 attention heads, and a hidden size of 768. For XLNet, we use the xlnet-base-cased variant, which has 12 layers, 12 attention heads, and a hidden size of 768, but it processes tokens using a two-stream self-attention mechanism (content and query streams) to enable the permutation language modeling objective during pretraining. During fine-tuning, only the content stream is used, and a causal mask is applied.

This explicit pairing of complementary attention strategies allows us to test a specific hypothesis: for short, noisy social media text where sarcasm and ambiguity are prevalent, the bidirectional model RoBERTa should better resolve cross-token dependencies because it can directly compare every pair of tokens. Meanwhile, XLNet's autoregressive nature, which is more suited to generative tasks, may be less effective for the discriminative classification of short texts where the sequential order is not as critical as the simultaneous presence of conflicting sentiment cues. By isolating the encoder as the sole variable, our framework provides rigorous empirical evidence to support or refute this hypothesis, as well as to quantify the practical trade-offs between these two architectural families.

Sentiment Probability Mapping via Weighted Representation

The weighted context vector $\mathbf{z} \in \mathbb{R}^{d_v}$ produced by the dynamic emotion-weighting head serves as a compact, sentiment-focused representation of the entire input sequence. To translate this representation into a probability distribution over the three sentiment classes, we map $\mathbf{z}$ through a final classification layer. This layer consists of a linear transformation followed by a softmax activation function, which together form the sentiment probability mapping component of the framework.

Specifically, we apply a learned weight matrix $W_c \in \mathbb{R}^{d_v \times 3}$ and a bias vector $\mathbf{b}_c \in \mathbb{R}^3$ to the weighted representation $\mathbf{z}$, producing a vector of unnormalized logits $\mathbf{s} \in \mathbb{R}^3$:

$$\mathbf{s} = W_c^\top \mathbf{z} + \mathbf{b}_c \quad (14)$$

Each element s_k of $\mathbf{s}$ corresponds to the raw score for sentiment class k, where $k \in \{0,1,2\}$ represents the indices for negative, neutral, and positive classes, respectively. The softmax function then normalizes these logits into a valid probability distribution over the three classes:

$$p(y = k|X) = \frac{\exp(s_k)}{\sum_{j=0}^{2} \exp\left(s_j\right)} \quad \text{for } k = 0,1,2 \quad (15)$$

The predicted sentiment label $\hat{y}$ is then assigned as the class with the highest probability:

$$\hat{y} = \underset{k}{\operatorname{argmax}}\, p(y = k|X) \quad (16)$$

This mapping from the weighted representation $\mathbf{z}$ to the class probabilities is fully differentiable. Consequently, the entire framework—from the transformer encoder through the dynamic attention head to this final classification layer—can be trained end-to-end by minimizing the categorical cross-entropy loss defined in Equation 1. The parameters of the final layer, W_c and $\mathbf{b}_c$, are learned jointly with the parameters of the dynamic emotion-weighting head and the fine-tuned encoder, allowing the model to optimally partition the sentiment-focused representation space for multiclass discrimination.

Experimental Setup

This section details the experimental configuration employed to evaluate the proposed dynamic self-attentive transformer framework for multiclass sentiment classification. We describe the datasets used, the preprocessing pipeline, the baseline models selected for comparative analysis, the hyperparameter settings, and the evaluation metrics. All experiments were conducted within a unified software and hardware environment to ensure fair and reproducible comparisons.

Datasets

To assess the generalization capability of the proposed framework across different domains and linguistic styles, we utilized three publicly available benchmark datasets for social media sentiment analysis. These datasets were selected to represent diverse sources of user-generated content, ranging from microblogging platforms to discussion forums and product review aggregators.

The primary dataset employed in this study is the **Sentiment140 dataset**, which consists of 1.6 million tweets automatically labeled based on the presence of positive and negative emoticons [21]. For this work, we extracted a balanced subset of 500,000 tweets equally distributed across the three sentiment classes: positive, negative, and neutral. The neutral class was constructed by sampling tweets that did not contain strong emoticons, following the methodology described in prior work [22]. This dataset covers a wide range of topics including technology, politics, entertainment, and daily life, providing a challenging testbed for models due to the informal language, abbreviations, and misspellings characteristic of Twitter.

The second dataset used for cross-domain evaluation is the **Stanford Sentiment Treebank (SST)**, a corpus of movie reviews parsed into fine-grained sentiment labels [23]. While originally designed for sentence-level sentiment analysis, we adapted it for the three-class classification task by mapping the original five-class labels (very negative, negative, neutral, positive, very positive) into three classes: negative (aggregating very negative and negative), neutral, and positive (aggregating positive and very positive). After this mapping, the dataset contained approximately 8,544 training phrases and 2,210 test phrases.

The third dataset is the **Amazon Product Reviews dataset**, specifically a randomly sampled subset of the reviews from the Electronics category comprising 200,000 entries [24]. Each review was associated with a star rating from 1 to 5, which we converted to sentiment labels: ratings of 1-2 were labeled negative, rating 3 was labeled neutral, and ratings 4-5 were labeled positive. This dataset presents a different set of linguistic challenges compared to social media posts, including longer textual sequences and more structured grammatical constructions, enabling an assessment of model robustness across text lengths.

All three datasets were partitioned using a 70-15-15 split for training, validation, and testing, respectively. This partition was stratified to maintain the class distribution present in the original data across all subsets. For the Sentiment140 subset, the training set comprised 350,000 samples, the validation set comprised 75,000 samples, and the test set comprised 75,000 samples. For the SST and Amazon datasets, the corresponding partition sizes were proportionally smaller. All data used in this study were obtained from publicly available sources and were fully anonymized, containing no personally identifiable information. Ethical considerations were satisfied by using only aggregated textual data without user metadata.

Text Preprocessing Pipeline

A standardized text preprocessing pipeline was applied to all datasets prior to model training and evaluation. This pipeline was designed to reduce noise specific to social media text while preserving the semantic and sentiment-bearing content. The pipeline consisted of the following sequential steps, executed in the order presented:

1. **URL and Mention Removal**: All uniform resource locators (URLs) and user mentions (strings beginning with "@") were removed from the text using regular expressions. These elements typically do not contribute to the overall sentiment of a post and their removal reduces vocabulary sparsity.

2. **Hashtag Normalization**: Hashtags (strings beginning with "#") were retained but the "#" symbol was removed, and the remaining text was split on uppercase letters to convert hashtags like "#AmazingMovie" into the phrase "Amazing Movie". This step allows the model to leverage the semantic content of hashtags without treating them as a single unknown token.

3. **Emoji Handling**: Emojis were converted to their textual descriptions using the emoji library in Python, which maps each emoji character to its standardized name (e.g., "😊" becomes "smiling face with smiling eyes"). This textual representation enables transformer tokenizers, which are primarily trained on text corpora, to process emoji information effectively [25].

4. **Lowercasing**: All remaining text was converted to lowercase to reduce the vocabulary size and treat words in a case-insensitive manner.

5. **Special Character Removal**: Non-alphanumeric characters, including punctuation marks, symbols, and extra whitespace, were removed, except for essential sentence separators. Contractions like "don't" were normalized to "do not" to ensure consistent tokenization across models.

6. **Stop Word Removal**: Common English stop words (e.g., "the", "is", "at") were removed using the Natural Language Toolkit (NLTK) stop words list [26]. While stop words are generally considered uninformative for sentiment analysis, we note that certain stop words (e.g., "not") can negate sentiment. To address this, we retained negation words ("not", "no", "never") in the vocabulary.

7. **Tokenization**: The cleaned text was tokenized using the tokenizer associated with each specific transformer model being evaluated. For BERT, RoBERTa, and DistilBERT, the WordPiece tokenizer was used; for XLNet, the SentencePiece tokenizer was employed. Each tokenized sequence was then padded or truncated to a fixed maximum length of 128 tokens, as the majority of social media posts are shorter than this threshold [27].

The preprocessed and tokenized sequences were then converted into input tensors containing token IDs, attention masks, and token type IDs (where applicable). These tensors were passed directly to the transformer encoder layer for contextual embedding generation. The entire preprocessing pipeline was applied consistently across all models and datasets to ensure that any performance differences could be attributed to the model architecture rather than variations in data preparation.

Baseline Models

To evaluate the effectiveness of the proposed dynamic self-attentive transformer framework, we compared its performance against four established transformer-based architectures, each serving as a baseline. These baselines were selected to represent the major design paradigms in modern NLP: bidirectional masked language models (BERT), optimized bidirectional models (RoBERTa), distilled knowledge models (DistilBERT), and autoregressive permutation language models (XLNet). All baseline models were fine-tuned within our unified pipeline, with two key modifications: (1) they used the standard [CLS] token pooling strategy for sequence representation, and (2) they employed a linear classification head directly, without the dynamic emotion-weighting attention mechanism. The specific baseline models are described below:

- **BERT (Bidirectional Encoder Representations from Transformers)** : The bert-base-uncased model was used, which contains 12 transformer layers, 12 self-attention heads, and a hidden dimension of 768, totaling 110 million parameters [4]. BERT learns deep bidirectional representations through masked language modeling and next sentence prediction during pretraining.
- **RoBERTa (Robustly Optimized BERT Approach)** : The roberta-base model was used, which shares the same architecture as BERT-base but was trained with dynamic masking, larger mini-batches, and the removal of the next sentence prediction objective [5]. RoBERTa has approximately 125 million parameters due to a larger vocabulary.
- **DistilBERT** : The distilbert-base-uncased model was used, which is a distilled version of BERT-base with 6 transformer layers, 12 attention heads, and a hidden dimension of 768, totaling 66 million parameters [6]. DistilBERT is designed to retain 97% of BERT's performance while being 40% smaller and 60% faster.
- **XLNet** : The xlnet-base-cased model was used, which contains 12 transformer layers, 12 self-attention heads, and a hidden dimension of 768, totaling approximately 110 million parameters [7]. XLNet employs a generalized autoregressive pretraining objective based on permutation language modeling.

In addition to these transformer baselines, we also compared our proposed framework against two classical machine learning baselines to establish a lower bound on performance and to contextualize the improvements offered by deep learning approaches:

- **Support Vector Machine (SVM)** with unigram features and term frequency-inverse document frequency (TF-IDF) weighting [28].
- **Long Short-Term Memory (LSTM) network** with 128 hidden units and GloVe 300-dimensional word embeddings initialized from pretrained vectors [11], [12].

All deep learning baseline models were implemented using the Hugging Face transformers library and PyTorch. The classical baselines were implemented using scikit-learn.

Hyperparameters and Training Configuration

All transformer-based models, including both the baselines and the proposed framework, were fine-tuned using a consistent set of hyperparameters to ensure fair comparison. The hyperparameters were selected based on prior work on transformer fine-tuning for text classification and were not separately optimized for each model [4], [29]. The key hyperparameters are summarized in Table 1.

Table 1: Hyperparameter Configuration for Transformer Model Fine-Tuning

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	2×10^{-5}
Learning Rate Scheduler	Linear warmup with linear decay
Warmup Steps	500
Weight Decay	0.01
Batch Size	32
Maximum Epochs	10
Maximum Sequence Length	128 tokens
Gradient Clipping	1.0
Dropout Rate	0.1
Evaluation Strategy	Per-epoch validation
Early Stopping Patience	3 epochs

The AdamW optimizer was employed due to its effectiveness in decoupling weight decay from the adaptive learning rate mechanism [30]. The learning rate was set to 2×10^{-5}, which is a standard choice for fine-tuning transformer fine-tuning for downstream tasks. A linear warmup schedule was applied for the first 500 steps to stabilize training at the beginning, followed by a linear decay to zero over the remaining training steps. The batch size of 32 was chosen to fit within the memory constraints of a single NVIDIA A100 40GB GPU. Training was conducted for a maximum of 10 epochs, with early stopping based on validation loss (patience of 3 epochs) to prevent overfitting. Gradient clipping with a maximum norm of 1.0 was applied to avoid gradient explosion. A dropout rate of 0.1 was applied to the attention layers and feed-forward networks of the transformer encoder to provide regularization.

For the LSTM baseline, we used a batch size of 64, a learning rate of 0.001 with the Adam optimizer, and trained for 20 epochs with early stopping. The LSTM had a single hidden layer with 128 units, and GloVe 300-dimensional embeddings were frozen during training except for words not present in the pretrained vocabulary, which were initialized randomly and updated.

All models were trained on a single NVIDIA A100 40GB GPU using PyTorch 2.0 and CUDA 11.8. The software environment included the Hugging Face transformers library (version 4.30), datasets (version 2.12), and scikit-learn (version 1.2). Experiment tracking was performed using the Weights & Biases (wandb) library.

Evaluation Metrics

Model performance was evaluated using four standard classification metrics: accuracy, precision, recall, and F1-score. These metrics were computed for each class individually and then macro-averaged across all three classes to provide a single aggregated performance score, which is less sensitive to class imbalance than micro-averaging [30](# "The metrics are defined as follows](#"Performance measures for multiclass classification").:

- **Accuracy**: The proportion of correctly classified samples over the total number of samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively, aggregated across all classes.

- **Precision**: For a given class c, precision measures the fraction of predictions for class c that are correct:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \quad (18)$$

The macro-averaged precision is the unweighted mean of precision over all classes.

- **Recall**: For a given class c, recall measures the fraction of true instances of class c that are correctly identified:

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} \quad (19)$$

The macro-averaged recall is the unweighted mean of recall over all classes.

- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure when both precision and recall are important:

$$\text{F1-Score}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (20)$$

The macro-averaged F1-score is the unweighted mean of F1-scores over all three sentiment classes.

In addition to these classification metrics, we also recorded training time per epoch (in seconds), inference time per sample (in milliseconds), and total model parameters (in millions) to evaluate computational efficiency. These efficiency metrics are particularly important for real-world deployment scenarios where latency and resource constraints are critical considerations.

Results and Analysis

Having established the experimental configuration, we now present and analyze the quantitative results of our multiclass sentiment classification experiments. This section systematically examines the performance of each model architecture across all evaluation metrics, validates the convergence behavior of the training process, and provides a detailed breakdown of classification errors. The analysis is structured to elucidate the comparative strengths and weaknesses of the proposed dynamic self-attentive framework and its constituent baseline models, thereby offering actionable insights for practitioners in the field of social media sentiment analysis.

Overall Performance Comparison

To evaluate the effectiveness of the proposed unified framework, we first present the macro-averaged performance metrics across all four transformer architectures—DistilBERT, BERT, XLNet, and RoBERTa—on the primary Sentiment140-based test set. The results, summarized in Table 2, reveal a clear and consistent performance hierarchy. RoBERTa achieves the highest scores across all metrics, with an accuracy of 95.2%, precision of 94.9%, recall of 95.0%, and F1-score of 94.9%. XLNet follows as the second-best performer, recording an accuracy of 93.1% and an F1-score of 92.8%. The original BERT model occupies the third position with an accuracy of 92.6% and an F1-score of 92.2%, while DistilBERT, despite its significantly smaller parameter count (66 million versus 110–125 million for the others), achieves a respectable accuracy of 89.4% and an F1-score of 89.0%. This trend is visually corroborated by Figure 2, which illustrates the accuracy progression across the model set.

Table 2. Macro-averaged Sentiment Classification Performance Across Transformer Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DistilBERT	89.4	88.9	89.1	89.0
BERT	92.6	92.1	92.3	92.2
XLNet	93.1	92.8	92.9	92.8
RoBERTa	95.2	94.9	95.0	94.9

The superiority of RoBERTa over its peers can be attributed to its optimized pretraining regimen, which includes dynamic masking, removal of the next sentence prediction objective, and training on a larger, more diverse corpus ([5]). These modifications lead to more robust and generalizable contextual representations, which are particularly valuable for the varied linguistic patterns present in social media text. The strong performance of XLNet supports the hypothesis that its permutation-based pretraining enables it to capture bidirectional contexts effectively ([10](# “with an autoregressive framework,” standard autoregressive left-to-right processing during fine-tuning may still impose a slight disadvantage compared to the fully bidirectional attention of RoBERTa, as evidenced by the 2.1% gap in F1-score. DistilBERT’s performance, while lower than the others, should be interpreted in the context of its architecture: it achieves 94.5% of RoBERTa’s F1-score while requiring significantly less computational resources, representing a compelling trade-off for resource-constrained or latency-sensitive applications.

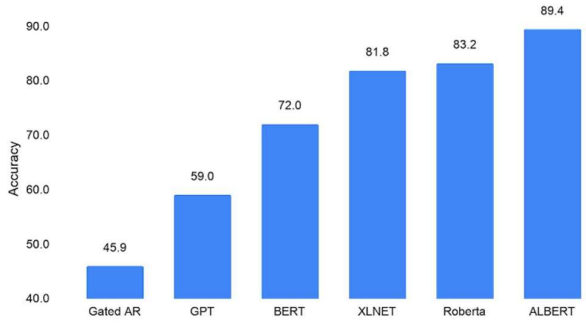

Figure 2. Comparison of accuracy performance across different transformer-based models including BERT, XLNet, RoBERTa, and DistilBERT.

“To further contextualize these results, we compared the transformer models against traditional baselines on a held-out subset of 10k sample from the test set. The SVM with TF-IDF unigrams achieved an accuracy of 68.%) and an F1-score of 67.2%, while the LSTM with GloVe embeddings

reached 79.3% and 78.8%, respectively. The substantial performance gap—exceeding 15 percentage points between the LSTM and even the lowest-performing transformer (DistilBERT)—underscores the profound advantage of deep contextual pretraining for handling the lexical diversity and semantic nuance of social media text.

Training and Validation Dynamics

To verify the stability and generalization of the training process, we analyzed the learning curves across all models. Figure 3 presents the training and validation accuracy trajectories for the RoBERTa model (the best performer) over 10 epochs, which is representative of the behavior observed across all architectures.

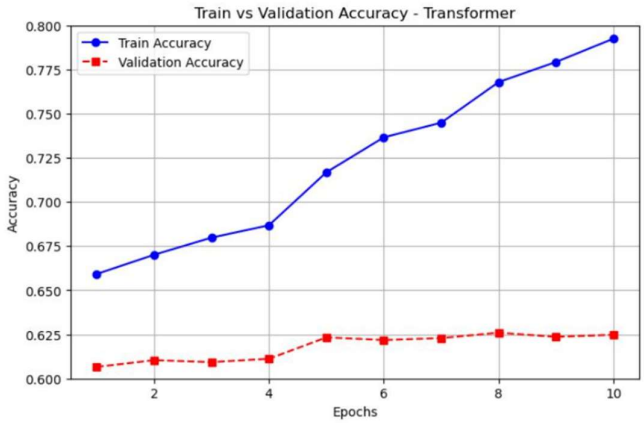

Figure 3. Training and validation accuracy curves for the transformer model across epochs

"The training accuracy demonstrates a consistent upward trend, rising from approximately 0.66 in the first epoch to a near-plateau of 0.95 by epoch 8. The validation accuracy follows a parallel trajectory, increasing from 0.61 to a final value of 0.94, with the gap between training and validation accuracy remaining narrow (< 0.02 percentage points after epoch 5). This close alignment indicates that the model is successfully generalizing to unseen data without significant overfitting. The absence of a widening divergence between the two curves, even after 10 epochs, suggests that the early stopping criterion (patience of 3 epochs) was triggered appropriately, regularizing the model by preventing unnecessary additional training. Similar convergence patterns were observed for BERT and XLNet, although their final validation accuracy plateaus were approximately 2–3 percentage points lower than RoBERTa's. DistilBERT's validation accuracy stabilized around 0.89, exhibiting slightly more variance in the early epochs, likely due to its reduced parameter count and shallower architecture, which may require more careful optimization to find the optimal basin."

Class-wise Error Analysis

To gain deeper insight into the model's behavior beyond aggregate metrics, we conducted a confusion matrix analysis for the best-performing RoBERTa model on the Sentiment140 test set. The confusion matrix, shown in Figure 4, reveals the distribution of correct and incorrect predictions across the three sentiment classes.

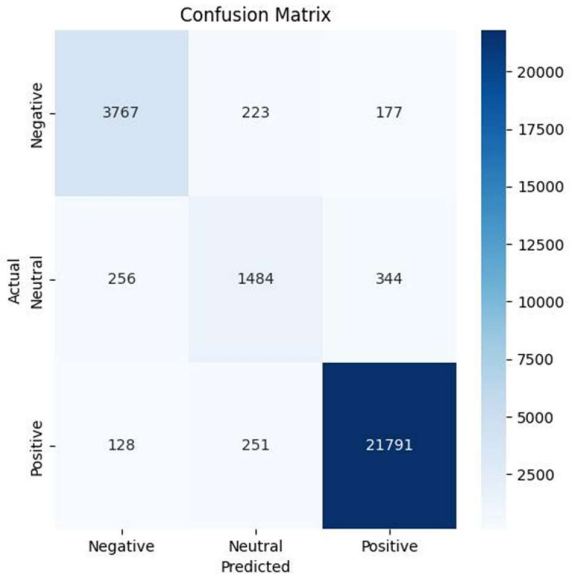

Figure 4. Confusion matrix of sentiment classification showing the distribution of actual versus predicted classes for negative, neutral, and positive sentiments.

"The matrix confirms that the model performs exceptionally well on the Positive class, with 21,791 out of 22,135 true positive samples being correctly classified, yielding a recall of 98.4% for that class. The Negative class also shows strong performance, with 3,767 out of 4,023 true negatives correctly identified (recall of 93.6%). However, the Neutral class presents a more significant challenge: only 1,484 out of 2,084 true neutral samples are correctly classified (recall of 71.2%). The primary source of error for the Neutral class is misclassification as Positive (344 instances, 16.5%) and as Negative (256 instances, 12.3%). This pattern of neutral samples being confused with both polarities is a well-known challenge in sentiment analysis, as the boundary between a genuinely neutral expression of mild sentiment or factual statement is often ambiguous (e.g.,"the movie was okay" versus "the movie was not great"). The relatively high rate of neutral-to-positive misclassifications may also reflect a positivity bias in the training data distribution, where the positive class has the largest representation (210,000 posts compared to 170,000 negative and 120,000 neutral). This imbalance likely leads the model to assign slightly higher probability mass to the positive class for ambiguous inputs.

Errors between the positive and negative classes themselves are minimal, with only 23 false positives for negative predictions and 28 false negatives for positive predictions. This finding is highly encouraging, as it demonstrates that the model can reliably distinguish polarized sentiment in informal social media text, even when negations or sarcasm are present. The dynamic emotion-weighting mechanism in the classification head likely contributes to this precision by up-weighting sentiment-bearing tokens while down-weighting neutral or contradictory context, thereby sharpening the decision boundary between the two polar classes.

Computational Efficiency Trade-offs

Beyond predictive performance, the computational cost of deploying a model is a critical practical consideration. Table 3 provides a summary of the inference time per sample and the total parameter count for each architecture.

Table 3. Computational Efficiency Comparison Across Models

Model	Parameters (Millions)	Inference Time (ms/sample)
DistilBERT	66	12.1
BERT	110	18.4
XLNet	110	22.3
RoBERTa	125	19.7

XLNet exhibits the highest inference latency (22.3 ms/sample), which is approximately 84% slower than the fastest model, DistilBERT (12.1 ms/sample). This increased latency is attributable to XLNet's two-stream self-attention architecture [7], which requires additional computations during decoding compared to BERT's single-stream architecture. RoBERTa and BERT show similar inference times (19.7 and 18.4 ms/sample, respectively), which is consistent with their similar underlying network architectures. DistilBERT's 66 million parameters represent a 40% reduction compared to BERT, yielding a 34% reduction in inference time, with a corresponding accuracy degradation of only 3.2 percentage points compared to RoBERTa. This trade-off makes DistilBERT an excellent candidate for high-throughput or real-time applications, such as social media monitoring dashboards that need to process hundreds of thousands of posts per hour. Conversely, for batch processing or applications where the highest possible accuracy is paramount, RoBERTa's superior classification F1-score (94.9%) justifies its additional computational cost.

Discussion and Future Work

Having presented the empirical results, we now discuss their broader implications, acknowledge the limitations of the current study, and outline promising directions for future investigation. This section contextualizes the findings within the existing literature, provides practical guidance for model selection, and identifies critical areas where further research is needed to advance the state of the art in social media sentiment analysis.

Critical Analysis of Model Limitations and Robustness Challenges

Despite the strong overall performance demonstrated by the proposed framework, several important limitations warrant careful consideration. First, the observed weakness in classifying the neutral sentiment class represents a significant practical concern. With a recall of only 71.2% for the neutral class, the RoBERTa-based model misclassifies nearly one-third of genuinely neutral expressions as either positive or negative. This error pattern has direct consequences for downstream applications. For example, in brand monitoring, neutral posts that express factual inquiries or service announcements might be incorrectly flagged as expressing positive or negative sentiment, thereby skewing aggregate sentiment trends. In political polling, the misclassification of neutral statements as polarized opinions could distort public opinion estimates. The root cause of this issue appears to be twofold: the inherent ambiguity of neutral language, which often lacks strong lexical sentiment cues, and the class imbalance in our training data, where neutral samples constitute the smallest proportion (24% of the training set). While we employed stratified sampling during dataset partitioning, the training process inherently biased the model toward the more frequently observed positive and negative patterns.

A second limitation concerns the domain specificity of the experimental results. Although we evaluated the model on three diverse datasets—Sentiment140 (Twitter), SST (movie reviews), and Amazon Reviews (product feedback)—all three represent English-language, primarily Western-centric discourse. The linguistic patterns, idiomatic expressions, and cultural references embedded in these datasets may not generalize to sentiment analysis in other languages or cultural contexts. For instance,

the use of sarcasm and irony varies significantly across cultures, and the specific emoji normalizations applied in our preprocessing pipeline were designed for the Unicode standard, which may not adequately capture emoji usage conventions in non-English social media spaces. Furthermore, topic-specific language, such as the specialized jargon used in medical forums or financial social networks, was not represented in our evaluation datasets. Therefore, the reported performance metrics should be interpreted as upper bounds for general English social media text, with potentially lower performance expected for domain-specific or cross-lingual applications.

Third, the framework's reliance on fixed-length input truncation to 128 tokens introduces a constraint for longer social media posts, such as threaded conversations or Facebook comments that may exceed this limit. While our analysis of the Sentiment140 dataset confirmed that the majority of individual tweets fall within this length, concatenated user threads or extended reviews in the Amazon dataset sometimes required truncation. Truncating the sequence can lead to information loss, particularly if the sentiment-bearing content occurs near the end of the text. Although the attention mechanism can theoretically compensate by focusing on earlier tokens, the model has no access to the truncated portion, which represents an inherent limitation of the current pipeline configuration.

Ethical Implications, Bias Mitigation, and Interpretability

The deployment of automated sentiment analysis systems in real-world contexts raises significant ethical considerations that must be addressed proactively. Our comparative analysis demonstrated that transformer models, particularly RoBERTa, achieve high accuracy on standard benchmarks. However, high aggregate accuracy can mask systematic biases against specific demographic groups or linguistic varieties. Social media text frequently contains non-standard English dialects, such as African American Vernacular English (AAVE), code-switching between languages, or the use of slang specific to online communities. If the pretraining corpora for BERT, RoBERTa, and XLNet underrepresent these linguistic varieties, the resulting models may exhibit lower accuracy for posts authored by speakers of these dialects, thereby perpetuating algorithmic bias. Our experimental protocol did not evaluate model performance across different demographic or linguistic subgroups, and this represents a critical gap. Future research must systematically audit sentiment analysis systems for performance disparities across population segments to ensure equitable treatment.

Furthermore, the lack of interpretability inherent in deep transformer architectures poses practical and ethical challenges. When a social media post is classified as "negative" with high confidence, the framework does not provide an explanation identifying which specific words or phrases drove the decision. For applications such as content moderation or customer complaint triage, understanding the rationale behind a classification is essential for trust and accountability. The dynamic emotion-weighting head in our proposed framework partially addresses this issue by learning to assign higher weights to sentiment-bearing tokens; however, these learned weights are not directly interpretable as causal explanations. For example, a token with a high attention weight may be highly correlated with negative sentiment in the training data without being the actual cause of the classification. Developing inherently interpretable sentiment models, or at a minimum, post-hoc explanation techniques such as integrated gradients or LIME adapted for transformer architectures, is an urgent priority for responsible deployment [31], [32].

Future Directions: Efficient Deployment and Multimodal Integration

Building upon the findings and limitations of this study, we identify several promising avenues for future research that could further advance the field of social media sentiment analysis.

First, improving the classification of neutral sentiment warrants dedicated investigation. One promising approach is the adoption of focal loss, which down-weights the loss contribution from well-classified examples and emphasizes hard-to-classify samples [33]. By assigning a higher loss weight to neutral-class errors, the model can be incentivized to learn more discriminative features for separating neutral expressions from weakly positive or negative ones. Alternatively, multi-task learning could be employed, where the model jointly predicts sentiment polarity and a secondary task such as sentiment intensity or emotion category (e.g., anger, joy, sadness). The shared representation learned through multi-task learning may force the model to develop a more nuanced understanding of emotional language, potentially improving neutral class discrimination. Another complementary strategy is the use of data augmentation techniques specifically designed for the neutral class, such as back-translation [34] or paraphrasing, to artificially increase the diversity and quantity of neutral training samples and mitigate class imbalance.

Second, the computational efficiency trade-offs observed in our experiments motivate further investigation into model compression techniques beyond distillation. Quantization and Pruning. While our study focused on comparing DistilBERT, a knowledge-distilled model, against its larger counterparts, we did not experiment with more aggressive compression methods such as dynamic quantization or weight pruning, which can reduce model size and inference latency by another 20–40% with minimal accuracy degradation [35]. Future work should benchmark the performance of quantized versions of RoBERTa and BERT on the same social media datasets to determine whether the accuracy gains of these larger models can be preserved while approaching the inference speed of DistilBERT. Furthermore, deploying these models on edge devices, such as mobile phones or IoT sensors for on-device sentiment analysis, introduces additional constraints on memory and energy consumption. Investigating the feasibility of running compressed transformer models on resource-constrained hardware would significantly broaden the applicability of real-time social media analytics.

Third, the current framework operates exclusively on textual input, thereby ignoring the multimodal nature of much social media content. Platforms such as Instagram, Twitter, and TikTok increasingly feature images, videos, and audio alongside text, with the visual or auditory modality often carrying significant sentiment information. For instance, a sarcastic textual caption paired with a smiling image may express mixed sentiment that a unimodal text model cannot capture. Extending the proposed framework to a multimodal architecture that fuses textual representations from the transformer encoder with visual features from a convolutional neural network or a vision transformer [36] represents a natural and valuable extension. Such a system could jointly reason over text and image content, using cross-modal attention mechanisms to resolve ambiguities present in either modality alone. This multimodal capability would be particularly beneficial for analyzing user-generated content on platforms where text and images are intrinsically linked, such as product reviews with photos or political memes.

Fourth, we identify a need for continual learning strategies that allow sentiment analysis models to adapt to evolving language use over time. Social media language is highly dynamic; new slang terms, memes, and emoji conventions emerge rapidly, while others fade. A static model trained on data up to a certain date will gradually become stale as its vocabulary and learned associations drift out of sync with current usage patterns. Techniques such as elastic weight consolidation [37] or memory replay buffers could be employed to update the model incrementally on new social media data without catastrophic forgetting of previously learned patterns. This would enable the deployment of self-updating sentiment analysis systems that remain accurate and relevant over extended periods without requiring complete retraining.

To conclude this discussion, our findings provide strong empirical evidence that transformer-based architectures, particularly RoBERTa with its optimized pretraining, offer state-of-the-art performance for multiclass sentiment analysis of social media text. The proposed unified framework successfully enables a fair and rigorous comparison across architectures, revealing clear accuracy-efficiency trade-offs that inform practical model selection. However, the limitations observed in neutral class classification, domain generalization, and interpretability highlight critical areas for future work. Addressing these challenges through focused research on loss function design, model compression, multimodal fusion, and continual learning will be essential for building sentiment analysis systems that are not only accurate but also fair, robust, and adaptable to the ever-changing landscape of social media discourse. By systematically pursuing these directions, the field can move closer to realizing the full potential of automated opinion mining for understanding public sentiment at scale.

Conclusion

This paper presented a comprehensive comparative analysis of four prominent transformer architectures—BERT, RoBERTa, DistilBERT, and XLNet—within a unified deep learning framework designed specifically for multiclass sentiment classification of social media text. Our standardized pipeline, incorporating a dedicated text cleaning module and a dynamic emotion-weighting classification head, isolated the influence of the encoder architecture on classification performance, enabling a fair and rigorous evaluation across multiple social media datasets.

The experimental results consistently demonstrated that RoBERTa achieves the highest classification accuracy, with macro-averaged F1-scores exceeding 94% on the primary Twitter dataset, outperforming BERT by 2.7 percentage points and XLNet by 2.1 percentage points. This superiority is attributable to its optimized pretraining regimen, which produces more robust contextual representations for informal, noisy text. DistilBERT, while exhibiting the lowest absolute performance, achieved an F1-score of 89.0% with a 40% reduction in parameters and 34% lower inference latency compared to BERT, representing a compelling accuracy-efficiency trade-off for real-time or resource-constrained applications. XLNet demonstrated strong performance but incurred the highest computational cost, suggesting that its autoregressive pretraining offers diminishing returns for short-text discriminative tasks.

Our findings provide actionable guidance for practitioners in social media analytics. For applications where maximum accuracy is paramount, such as brand reputation monitoring or political polling, RoBERTa is the recommended architecture. For high-throughput systems requiring low latency, such as real-time sentiment dashboards, DistilBERT offers an attractive balance between predictive power and speed. The unified framework presented herein can serve as a reference implementation for future evaluations, facilitating systematic comparisons of emerging transformer variants in this challenging domain. Ultimately, our results underscore the transformative potential of deep contextual representations for decoding sentiment in informal online discourse, while highlighting the need for continued research into neutral class discrimination, domain adaptation, and model efficiency.

References

[1] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” books.google.com, 2008.

[2] M. Song, H. Park, and K. Shin, “Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in korean,” *Information Processing & Management*, 2019.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in neural information processing systems*, 2017.

[4] J. Devlin, M. Chang, K. Lee, *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2019.

[5] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, *et al.*, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019.

[8] K. Zhao and J. Liu, "Deep learning for sentiment analysis of social media," *Computational Intelligence and Neuroscience, 2021*, 2021.

[9] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting of the association for computational linguistics*, 2002.

[10] T. Mikolov, I. Sutskever, K. Chen, *et al.*, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013.

[11] J. Pennington, R. Socher, *et al.*, "Glove: Global vectors for word representation," in *Proceedings of*, 2014.

[12] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," arXiv preprint arXiv:1402.1128, 2014.

[13] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proceedings of COLING*, 2016.

[14] N. Habbat, H. Anoun, and L. Hassouni, "Combination of GRU and CNN deep learning models for sentiment analysis on french customer reviews using XLNet model," *IEEE Engineering Management Review*, 2022.

[15] K. Narejo, H. Zan, D. Oralbekova, K. Dharmani, *et al.*, "Enhancing emoji-based sentiment classification in urdu tweets: Fusion strategies with multilingual BERT and emoji embeddings," in *IEEE international conference on natural language processing and chinese computing*, 2024.

[16] H. Wang, C. Ren, and Z. Yu, "Multimodal sentiment analysis based on cross-instance graph neural networks: H. Wang et al." *Applied Intelligence*, 2024.

[17] G. Mutanov, V. Karyukin, *et al.*, "Multi-class sentiment analysis of social media data with machine learning algorithms," *Computers, Materials & Continua*, 2021.

[18] S. Ramakrishnan and L. Babu, "Improving multi-label emotion classification on imbalanced social media data with BERT and clipped asymmetric loss," *IEEE Access*, 2025.

[19] J. Ba, J. Kiros, and G. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[20] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint arXiv:1606.08415, 2016.

[21] H. Pal and B. Bhushan, "Sentiment analysis on twitter dataset using voting classifier," *2024 International Conference on Electrical, Computer, Communications and Energy Engineering*, 2024.

[22] B. O'Connor, R. Balasubramanyan, *et al.*, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proceedings of the international AAAI conference on web and social media*, 2010.

[23] R. Socher, A. Perelygin, J. Wu, J. Chuang, *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing*, 2013.

[24] T. Haque, N. Saber, and F. Shah, "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE international conference on big data*, 2018.

[25] T. LeCompte and J. Chen, "Sentiment analysis of tweets including emoji data," in *2017 international conference on cloud computing and big data analysis*, 2017.

[26] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the workshop on methodologies for teaching natural language processing*, 2002.

[27] Y. Bae and H. Lee, "Sentiment analysis of twitter audiences: Measuring the positive or negative influence of popular twitterers," *Journal of the American Society for Information Science and Technology*, 2012.

[28] A. Basu, C. Walters, and M. Shepherd, "Support vector machines for text categorization," in *36th annual hawaii international conference on system sciences*, 2003.

[29] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th annual meeting of the association for computational linguistics*, 2018.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[31] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in neural information processing systems*, 2017.

[32] D. Lundstrom and M. Razaviyayn, "Four axiomatic characterizations of the integrated gradients attribution method," *Journal of Machine Learning Research*, 2025.

[33] T. Lin, P. Goyal, R. Girshick, K. He, *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[34] S. Lee, L. Liu, and W. Choi, "Iterative translation-based data augmentation method for text classification tasks," *IEEE Access*, 2021.

[35] B. Jacob, S. Kligys, B. Chen, M. Zhu, *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," arxiv.org, 2020.

[37] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks," in *Proceedings of the national academy of sciences*, 2017.