Digital Object Identifier 10.48165/iitmjit.2026.12.1.5

An Attention-Enhanced DeBERTa BiLSTM Hybrid Architecture for Robust Multi-Class Sentiment Classification

Mansi Bansal[1], Saurabh Sharma[2], Sonal Singh[3], Priyanka Singh[4]
[1,4]Assistant Professor, [2, 3] Research Scholar, [1]Department of Computer Science, [3]Department of Information Technology, [2,4]School of Computer Science Engineering and Technology
[1]GGSIPU, Delhi, India, [2]Bennett University, Greater Noida, Uttar Pradesh, India, Government Engineering College, Bilaspur, Chhattisgarh, India, Rishihood University, Bahalgarh, Sonipat, Haryana, India
[1]bansalmansi2@gmail.com, [2]Saurabh180704@gmail.com, [3]singhsonals999@gmail.com, [4]priyankasngh99@gmail.com

Abstract: The widespread sharing of user-generated text on social media, review platforms, and online forums has intensified the demand for accurate, scalable, and context-aware sentiment analysis. Despite significant advancements in deep learning, sentiment classification remains challenging due to contextual ambiguity, linguistic diversity, domain dependency, long-distance semantic relationships, and class imbalance in real-world datasets. Traditional machine learning approaches rely heavily on handcrafted features and shallow representations, which limit their ability to capture deep semantic context. Although transformer-based models provide strong contextual embeddings, standalone transformers may not fully exploit sequential temporal dependencies that are essential for fine-grained sentiment modelling.

In this work, we present an enhanced hybrid deep learning framework that extends existing transformer–recurrent architectures by integrating a pre-trained DeBERTa-v3-base encoder with a stacked BiLSTM network and an explicit additive attention mechanism for multi-class sentiment classification. The DeBERTa encoder leverages disentangled attention and large-scale pretraining to generate rich contextualized representations, effectively modelling complex semantic relationships. The BiLSTM layer further refines these embeddings by capturing bidirectional sequential dependencies, while the attention layer selectively emphasizes sentiment-discriminative tokens within the text. To address dataset imbalance and improve lexical diversity, word-embedding-based data augmentation is employed.

The proposed architecture is evaluated on widely used benchmark datasets including IMDb, Twitter US Airline, and Sentiment140, and is compared against classical machine learning models, standalone deep learning approaches, and transformer–recurrent hybrid baselines. Experimental results demonstrate consistent improvements in accuracy, precision, recall, and F1-score, particularly on short-form social media text. The proposed framework is scalable and generalizable, offering a robust solution for real-world applications in business intelligence, social analytics, and decision-support systems.

Keywords: Sentiment analysis, DeBERTa-v3-base, Bidirectional Long Short-Term Memory (BiLSTM), attention mechanism, Transformer, hybrid deep learning, text classification.

1. Introduction

The rapid increase of user-generated text on social media, review websites, and discussion forums has made sentiment analysis (SA) a fundamental natural language processing (NLP) task. Sentiment analysis (SA) is the automatic classification of sentiments expressed in a text, typically into positive, negative, or neutral. It is important in many brand monitoring applications, business intelligence, political opinion mining, healthcare analysis, and decision-support systems [1], [2], [17]. The growing volume, variety, and velocity of textual data have intensified the demand for accurate, scalable, and context-sensitive sentiment classifiers.

Though there have been many advancements in this area, sentiment classification remains a challenging problem. Real-world textual data usually contain informal language, lexical variation, sarcasm, domain dependency, and contextual ambiguity. Sentiment expressions often rely on long-range dependencies across words and sentences, which are difficult for traditional machine learning (ML) models to capture. Early ML

approaches relied on techniques such as Naïve Bayes, Support Vector Machines, and Logistic Regression. These methods depended on handcrafted features and shallow representations such as Bag-of-Words (BoW) and TF-IDF [15], [16]. While computationally efficient, these approaches fail to capture deeper semantic richness and contextual relationships within text.

To address these limitations, distributed word representation techniques such as Word2Vec [9] and GloVe [10] were introduced to encode semantic similarity between words in continuous vector spaces. These embeddings significantly improved downstream NLP tasks, including sentiment classification. However, static word embeddings assign a single representation to each word regardless of context, limiting their ability to capture polysemy and context-dependent sentiment. Consequently, sentiment understanding remained constrained at both sentence and document levels.

Subsequently, recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks, were introduced to model sequential text data [6], [7]. LSTMs effectively addressed the vanishing gradient problem and captured long-range dependencies, making them suitable for sentiment analysis. Convolutional neural networks (CNNs) were also applied to extract local n-gram features from sequential text [8]. Hybrid CNN-LSTM architectures further enhanced performance by combining local feature extraction with sequential modelling [14], [19]. However, RNN-based models process text sequentially, limiting parallelization and often struggling to efficiently capture global contextual relationships.

The Transformer architecture introduced parallel processing through the self-attention mechanism, revolutionizing NLP [5]. By attending to all tokens simultaneously, transformer-based architectures effectively model global contextual dependencies. BERT adopted this paradigm to generate bidirectional contextual embeddings and achieved strong performance across various NLP tasks [4]. RoBERTa further optimized BERT's pre-training strategy using larger datasets and dynamic masking [3], establishing itself as a powerful contextual encoder for sentiment analysis systems.

Nonetheless, although transformer architectures are highly effective at encoding global semantic information, standalone transformer models may not explicitly emphasize sequential temporal patterns within text. Recent studies suggest that integrating transformer-based contextual encoders with recurrent neural networks can leverage complementary strengths [1]–[3], [12], [13]. Tan et al. [3] proposed the RoBERTa-LSTM hybrid model, which combines RoBERTa embeddings with LSTM-based sequential modeling and demonstrated strong improvements over standalone deep learning models, establishing a competitive hybrid baseline.

Several transformer–recurrent hybrid frameworks have since been proposed. Semary et al. [1] explored RoBERTa combined with CNN and LSTM layers, while Jahin et al. [2] introduced an attention-enhanced transformer–BiLSTM architecture for robust and interpretable tweet sentiment analysis. Variants such as RoBERTa-GRU and RoBERTa-LSTM [12], [13] have shown competitive results across benchmark datasets. These findings indicate that hybrid architectures consistently outperform traditional ML and standalone deep learning approaches.

Despite these advancements, important challenges remain. Class imbalance in many sentiment datasets leads to biased predictions toward majority classes. Moreover, achieving robust generalization across domains, particularly for noisy and linguistically diverse social media text, remains difficult [2], [18]. Furthermore, prior RoBERTa-based hybrid models do not fully exploit disentangled attention mechanisms or explicit token-level weighting strategies that could further enhance sentiment discrimination.

Motivated by these observations and building upon the RoBERTa-LSTM framework proposed by Tan et al. [3], this study extends transformer–recurrent integration by incorporating a more advanced contextual encoder and an explicit attention mechanism. Specifically, we replace the RoBERTa encoder with DeBERTa-v3-base, which utilizes disentangled attention to model content and positional information more effectively. The proposed approach integrates DeBERTa-based contextual embeddings with stacked BiLSTM-based sequential modeling and an additive attention layer to capture global semantics, bidirectional temporal dependencies, and sentiment-discriminative token importance. Comprehensive evaluations are conducted on benchmark sentiment datasets, and the proposed model is compared against traditional ML approaches, standalone deep learning models, and RoBERTa-based hybrid baselines using standard evaluation metrics.

The main contributions of this work are summarized as follows:

- A robust hybrid sentiment analysis framework that integrates DeBERTa-v3-base contextual embeddings with stacked BiLSTM-based sequential modeling and an explicit additive attention mechanism.
- A comprehensive evaluation against state-of-the-art machine learning and deep learning baselines, including the RoBERTa-LSTM base model [3].
- An extensive performance analysis using Accuracy, Precision, Recall, and F1-score metrics to demonstrate effectiveness and generalization capability across multiple datasets.

The remainder of this paper is organized as follows. Section II reviews related work in sentiment analysis. Section III describes the proposed methodology and model architecture. Section IV presents the datasets and experimental setup. Section V discusses experimental results and analysis. Finally, Section VI concludes the paper and outlines future research directions.

2. Related work

Sentiment analysis (SA) has evolved significantly over the past decade, transitioning from traditional machine learning approaches to advanced deep learning and transformer-based architectures. Existing research can broadly be categorized into three major directions: (1) traditional machine learning-based methods, (2) deep learning-based architectures, and (3) hybrid transformer–recurrent frameworks.

2.1 Traditional Machine Learning Approaches

Early sentiment classification systems primarily relied on supervised machine learning algorithms using handcrafted textual features. Techniques such as Support Vector Machines (SVM), Naïve Bayes (NB), Decision Trees, and Logistic Regression were widely adopted for polarity detection tasks [15], [16]. These approaches typically used Bag-of-Words (BoW) and TF-IDF representations to convert textual data into numerical vectors. While computationally efficient and easy to implement, such shallow representations failed to capture contextual semantics and long-range dependencies in text.

Word embedding models such as Word2Vec [9] and GloVe [10] were introduced to overcome the limitations of sparse representations by mapping words into dense continuous vector spaces. These embeddings improved semantic representation by capturing syntactic and distributional similarity. However, they remained static in nature, assigning a single embedding to each word regardless of context, which limited their ability to handle polysemy and context-dependent sentiment expressions.

Although traditional ML models demonstrated reasonable performance in early benchmarks [15], [16], they lacked the capability to model deep contextual relationships, motivating the adoption of deep learning techniques.

2.2 Deep Learning-Based Sentiment Analysis

The introduction of deep neural networks marked a major shift in sentiment analysis research. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks were extensively applied due to their ability to model sequential dependencies in textual data [6], [7]. LSTM networks, in particular, addressed the vanishing gradient problem and effectively captured long-term contextual dependencies, making them suitable for sentence-level and document-level sentiment classification.

Convolutional Neural Networks (CNNs) were also successfully applied to text classification tasks, demonstrating strong performance in capturing local n-gram features [8]. Hybrid CNN–LSTM architectures further enhanced sentiment classification by combining local feature extraction with sequential modeling [14], [19]. These models outperformed traditional ML approaches by learning hierarchical representations directly from raw text data.

Despite these improvements, deep learning models based purely on RNNs or CNNs still faced limitations in capturing global contextual information efficiently. Sequential processing in RNNs restricted parallelization, and long sequences remained computationally expensive to handle.

2.3 Transformer-Based Architectures

The development of the Transformer architecture revolutionized NLP by introducing the self-attention mechanism, enabling parallel processing of sequences and improved modeling of global dependencies [5]. BERT (Bidirectional Encoder Representations from Transformers) significantly advanced contextual representation learning by pretraining deep bidirectional transformers on large-scale corpora [4]. RoBERTa further optimized BERT's training procedure, achieving improved robustness and performance through larger training datasets, dynamic masking, and hyperparameter refinements [3].

Transformer-based models have since demonstrated state-of-the-art results across numerous NLP tasks, including sentiment analysis [1], [2], [18]. Sentence-level embedding techniques such as Sentence-BERT extended contextual learning for improved semantic similarity modeling [11]. Moreover, multilingual transformer variants such as XLM-RoBERTa have been applied to cross-lingual sentiment classification [18], further enhancing generalization capabilities.

However, while transformer models excel at capturing global contextual semantics, they may not explicitly emphasize temporal sequence patterns that recurrent networks model naturally. This observation led to the emergence of hybrid architectures integrating transformer encoders with sequential neural networks.

2.4 Hybrid Transformer–Recurrent Architectures

Recent research has focused on combining the contextual representation power of transformers with the sequential modeling capability of recurrent neural networks. A seminal contribution in this direction is the RoBERTa-LSTM model proposed by Tan et al. [3], which integrates RoBERTa-based contextual embeddings with an LSTM layer for sequential refinement. Their model demonstrated significant improvements over standalone deep learning architectures on benchmark datasets, establishing a strong baseline for hybrid sentiment classification models.

Subsequent studies further expanded on this idea. Semary et al. [1] proposed a RoBERTa-based hybrid model incorporating CNN and LSTM layers, achieving improved classification accuracy on IMDb and Twitter datasets. Jahin et al. [2] introduced the TRABSA model, which combines RoBERTa with attention-based BiLSTM layers, emphasizing robustness and interpretability in tweet sentiment analysis. Similarly, hybrid RoBERTa-GRU and RoBERTa-LSTM frameworks have reported competitive results across multiple sentiment datasets [12], [13].

These hybrid architectures consistently outperform traditional ML models [15], [16] and standalone deep learning models [14], [19], demonstrating that transformer-recurrent integration effectively captures both global contextual information and sequential dependencies.

2.5 Research Gaps

Although hybrid transformer–recurrent models have shown promising results, several challenges remain. Many sentiment datasets suffer from class imbalance and domain variability, which impact model generalization [1], [20]. Additionally, achieving consistent performance across diverse social media datasets characterized by noisy and informal text remains an open research problem [2], [18].

Motivated by these limitations and inspired by the RoBERTa-LSTM baseline model [3], this study further investigates hybrid transformer–LSTM integration for robust multi-class sentiment classification, aiming to enhance contextual understanding, sequential modeling, and overall classification performance.

3. Proposed DeBERTa - BiLSTM - Attention Approach

In this section, we describe the proposed hybrid model, DeBERTa-BiLSTM-Attention, designed for sentiment analysis. The proposed model integrates the complementary strengths of Transformer-based and Recurrent Neural Network (RNN) architectures, further augmented by an explicit attention mechanism, to enhance efficacy and classification accuracy across diverse sentiment analysis benchmarks. The architecture of the proposed model is illustrated in Fig. 1. The pretrained DeBERTa-v3-base model serves as the foundational encoder in the proposed hybrid framework, tokenizing all input text and mapping tokens into rich, context-sensitive word embedding representations through its disentangled attention mechanism. These contextual word embeddings, generated by the pretrained DeBERTa encoder, are subsequently passed through a dropout layer before being fed into a two-layer Bidirectional Long Short-Term Memory (BiLSTM) network, which captures

long-range sequential dependencies within the embedding sequence in both forward and backward directions. Following the BiLSTM, a dedicated attention layer selectively weights the hidden states to focus on the most sentiment-discriminative regions of the input sequence, producing an enhanced context vector. A layer normalization step is then applied to stabilize training, after which a fully connected dense layer maps the attended representation to the output sentiment classes. Finally, a Softmax activation function estimates the probability distribution over the target classes. The overall steps of the proposed DeBERTa-BiLSTM-Attention hybrid model are outlined in Algorithm 1. The individual components of the proposed hybrid model are described in detail below.

3.1 DeBERTa

DeBERTa (Decoding-enhanced BERT with Disentangled Attention), introduced by He et al. [1], represents a significant advancement over conventional BERT-based pre-trained language models in the domain of natural language understanding (NLU). Unlike BERT [2] and RoBERTa [3], which encode each token using a single vector that jointly represents both the token content and its positional information, DeBERTa employs a disentangled attention mechanism that represents each token using two separate vectors — one encoding the token content and the other encoding the positional information. The attention weights between any two tokens are then computed using disentangled matrices that account for content-to-content, content-to-position, and position-to-content interactions. This architectural innovation enables DeBERTa to model the relative positions of tokens in a more fine-grained manner, significantly enhancing the model's ability to capture the nuanced semantic and syntactic relationships present in natural language text, which are critical for accurate sentiment classification.

The DeBERTa-v3-base variant adopted in the proposed model employs token detection (RTD) pre-training, inspired by ELECTRA [4], in conjunction with the gradient-disentangled embedding sharing (GDES) technique. These refinements yield a substantially more sample-efficient pre-training process compared to masked language modeling (MLM), enabling the model to achieve superior performance on downstream NLU tasks with less computational overhead. With its 12-layer Transformer architecture, 768-dimensional hidden states per layer, and 12 attention heads, DeBERTa-v3-base possesses 86 million parameters that encode rich contextual knowledge acquired from large-scale pre-training corpora. In the proposed hybrid model, the pretrained DeBERTa tokenizer decomposes raw input text into subword tokens using the SentencePiece tokenization scheme, thereby preserving semantic meaning while effectively handling out-of-vocabulary words and morphologically complex expressions. Each token is assigned a unique input identifier and an attention mask that facilitates focused encoding within the DeBERTa framework.

3.2 Dropout Layer

The dropout layer constitutes a fundamental regularization technique in deep learning models, playing a crucial role in preventing overfitting and promoting generalization to unseen data. Introduced by Srivastava et al. [5], dropout randomly deactivates a fraction of neurons within a layer during each forward pass of training, thereby reducing co-adaptation between neurons and discouraging the network from memorizing spurious patterns in the training data. This stochastic regularization technique has been widely and successfully adopted across a broad range of neural network architectures, including Convolutional Neural Networks (CNNs), RNNs, and Transformer-based models, yielding consistent improvements in generalization performance across tasks such as image classification, natural language processing, and speech recognition [6]. In the proposed DeBERTa-BiLSTM-Attention model, a dropout layer with a dropout rate of $d = 0.3$ is inserted between the DeBERTa encoder output and the BiLSTM layer input. This placement ensures that the contextual embeddings produced by

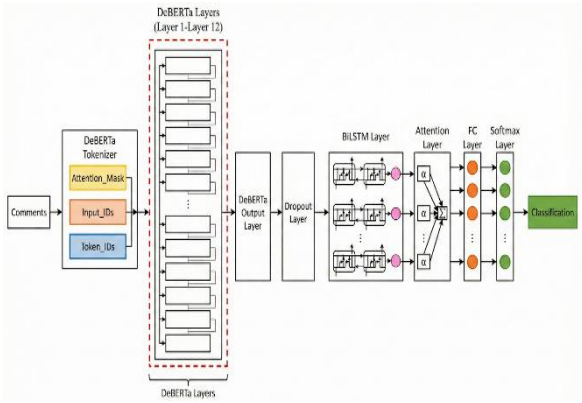

Fig. 5.1 Hybrid Deep learning architecture

DeBERTa are appropriately regularized prior to sequential processing by the BiLSTM network, thereby reducing the risk of overfitting that may arise from the high dimensionality of the DeBERTa output representations.

3.3 BiLSTM Layer

Bidirectional Long Short-Term Memory (BiLSTM), a specialized RNN architecture comprising two LSTM networks processing the input sequence in opposing temporal directions, plays a pivotal role in the proposed hybrid model. One LSTM network processes the input sequence from left to right (the forward LSTM, denoted →), while the other processes it from right to left (the backward LSTM, denoted ←) [7], [8]. This bidirectional processing paradigm enables the BiLSTM to capture rich contextual information from both past and future tokens simultaneously, rendering it particularly valuable for NLP tasks such as sentiment analysis [9], [10], named entity recognition [11], and machine translation. Hochreiter and Schmidhuber [12] introduced the foundational LSTM architecture to address the vanishing gradient problem that afflicts conventional RNNs when modeling long-range dependencies. Graves and Schmidhuber [7], [13] subsequently demonstrated the superiority of bidirectional LSTM models over their unidirectional counterparts in tasks requiring comprehensive contextual understanding. In the proposed model, a two-layer stacked BiLSTM architecture is employed, with each layer possessing $h = 256$ hidden units per direction, yielding 512-dimensional concatenated hidden states from the forward and backward passes. The two-layer design enables the model to capture hierarchical sequential features of progressively higher abstraction, enhancing its representational capacity compared to a single-layer BiLSTM. The BiLSTM model architecture is governed by the following set of equations [14], [15].

- Input Gate (i_t):

$$i\_t = \sigma(W^f\_ix \cdot x\_t + W^f\_ih \cdot h^f\_(t-1) + W^f\_ic \cdot c^f\_(t-1) + b^f\_i) \odot \sigma(W^b\_ix \cdot x\_t + W^b\_ih \cdot h^b\_(t+1) + W^b\_ic \cdot c^b\_(t+1) + b^b\_i) \quad (2)$$

Equation (2) controls the flow of new information into the cell state C_t at time step t in both the forward and backward directions, combining contributions from the respective LSTM networks using element-wise multiplication. The sigmoid activation function σ squashes values to the range [0, 1], determining the extent to which new input is incorporated into the cell state.

- Forget Gate (f_t):

$$f\_t = \sigma(W^f\_fx \cdot x\_t + W^f\_fh \cdot h^f\_(t-1) + W^f\_fc \cdot c^f\_(t-1) + b^f\_f) \odot \sigma(W^b\_fx \cdot x\_t + W^b\_fh \cdot h^b\_(t+1) + W^b\_fc \cdot c^b\_(t+1) + b^b\_f) \quad (3)$$

Equation (3) determines which information from the previous cell state c_{t-1} should be discarded or retained in both the forward and backward directions, combining the forget gate computations from both LSTM networks through element-wise multiplication.

- Cell State Update (C_t):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W^f_{cx} \cdot x_t + W^f_{ch} \cdot h^f_{t-1} + b^f_c) + i_t \odot \tanh(W^b_{cx} \cdot x_t + W^b_{ch} \cdot h^b_{t+1} + b^b_c) \quad (4)$$

Equation (4) updates the cell state C_t by combining retained information from the previous state with new candidate values computed from both forward and backward directions, weighted by the input gate activations.

- Output Gate (o_t):

$$o_t = \sigma(W^f_{ox} \cdot x_t + W^f_{oh} \cdot h^f_{t-1} + W^f_{oc} \cdot c^f_t + b^f_o) \odot \sigma(W^b_{ox} \cdot x_t + W^b_{oh} \cdot h^b_{t+1} + W^b_{oc} \cdot c^b_t + b^b_o) \quad (5)$$

- Hidden State (h_t):

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

Equations (5) and (6) together regulate which portions of the cell state C_t are exposed as the hidden state output h_t at each time step t in both processing directions. The concatenated bidirectional hidden states at each time step t are given by:

$$\rightarrow h_t \oplus \leftarrow h_t = BiLSTM(x_t, h_{t-1}) \quad (7)$$

where $\oplus$ denotes vector concatenation. The output of the two-layer BiLSTM is a sequence of 512-dimensional hidden state vectors $H_{BiLSTM} \in R^{l \times 2h}$, which encodes both local and long-range contextual dependencies across the entire input sequence, forming the input to the subsequent attention layer.

3.4 Attention Layer

The attention layer constitutes the key architectural contribution that distinguishes the proposed DeBERTa-BiLSTM-Attention model from the baseline RoBERTa-BiLSTM approach [16]. While the BiLSTM captures comprehensive sequential information across the entire input sequence, not all tokens contribute equally to the final sentiment classification decision. The attention mechanism addresses this limitation by learning to assign differential importance weights to each hidden state produced by the BiLSTM, enabling the model to focus selectively on the most sentiment-discriminative tokens and phrases within the input sequence. Bahdanau et al. [17] introduced this additive attention formulation in the context of neural machine translation, demonstrating that attention mechanisms substantially improve the ability of encoder-decoder architectures to process long sequences. In the proposed model, an additive attention mechanism is applied over the BiLSTM hidden state sequence to compute a context vector that serves as a weighted summary of the entire sequence for sentiment classification. The attention mechanism is formulated as follows:

- Attention Score (e_t):

$$e_t = v_a^T \cdot \tanh(W_a \cdot h_t + b_a) \quad (8)$$

where $h_t \in R^{2h}$ is the BiLSTM hidden state at time step t; $W_a \in R^{d_a \times 2h}$ is a learnable projection matrix; $b_a \in R^{d_a}$ is a bias vector; $v_a \in R^{d_a}$ is a learnable context vector; and d_a denotes the attention dimensionality. The scalar score e_t measures the relevance of the hidden state at position t to the final sentiment classification decision.

- Attention Weight (α_t):

$$\alpha_t = \exp(e_t) / \Sigma^l_{k=1} \exp(e_k) \quad (9)$$

Equation (9) applies a Softmax normalization over all attention scores to produce the attention weight distribution $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_l\}$, ensuring that the weights are non-negative and sum to unity. Tokens with

higher attention weights are deemed more informative for predicting the sentiment polarity of the input sequence.

- Context Vector (c):

$$c = \Sigma^{l}_{t=1} \alpha_t \cdot h_t \quad (10)$$

Equation (10) computes the context vector $c \in R^{2h}$ as a weighted sum of all BiLSTM hidden states, where the weights are the learned attention coefficients. This context vector captures a compact, attention-weighted representation of the entire input sequence that emphasizes the most sentiment-relevant token positions. Layer normalization [18] is subsequently applied to the context vector prior to classification, stabilizing the activation magnitudes and accelerating convergence:

$$c_{norm} = LayerNorm(c) = \gamma \cdot (c - \mu) / \sqrt{(\sigma^2 + \varepsilon)} + \beta \quad (11)$$

where μ and σ^2 are the mean and variance of the context vector elements; γ and β are learnable scale and shift parameters; and ε is a small constant added for numerical stability.

3.5 Dense Layer

The dense layer, also referred to as the fully connected layer, plays a crucial role in mapping the attended context representation to the sentiment class label space. This layer establishes dense connectivity by applying a learned linear transformation to the layer-normalized context vector, effectively capturing the non-linear relationships between the attended sequential features and the target sentiment classes. In the proposed model, a single dense layer is incorporated following the layer normalization step, projecting the 512-dimensional context vector onto a lower-dimensional representation aligned with the number of sentiment classes in the target dataset. A ReLU activation function is applied within this layer to introduce non-linearity, enabling the model to capture complex decision boundaries. The dense layer transformation is formulated as follows:

$$z = ReLU(W_d \cdot c_{norm} + b_d) \quad (12)$$

where W_d and b_d are the learnable weight matrix and bias vector of the dense layer respectively, and z is the resulting activation vector that serves as input to the final classification layer.

3.6 Classification Layer

The classification layer serves as the final output layer of the proposed DeBERTa-BiLSTM-Attention hybrid model. It applies the Softmax activation function to the output of the dense layer to generate a normalized probability distribution over the target sentiment classes, enabling the model to produce interpretable class probability estimates. The Softmax function is defined as follows:

$$Softmax(O)_j = e^{O_j} / \Sigma^{M}_{k=1} e^{O_k} \quad (13)$$

where M denotes the total number of sentiment classes in the target dataset; O_j represents the j-th element of the pre-softmax logit vector O; and the denominator $\Sigma^{M}_{k=1} e^{O_k}$ ensures that the output probabilities sum to unity across all classes. The predicted sentiment class $\hat{y}$ is assigned as the class with the highest predicted probability:

$$\hat{y} = argmax_j \; Softmax(O)_j \quad (14)$$

3.7 Loss Function and Optimization

Given that sentiment analysis constitutes a multi-class classification problem, label-smoothed cross-entropy is employed as the loss function L to optimize the parameters of the proposed model. Label smoothing introduces a regularization effect by preventing the model from becoming overconfident in its predictions, thereby improving generalization. The label-smoothed cross-entropy loss is formulated as follows:

$$L(p) = -\Sigma^{M}_{i=1} \tilde{y}_i \cdot log(\hat{p}_i) \quad (15)$$

where $\tilde{y}_i = (1 - \varepsilon) \cdot y_i + \varepsilon / M$ is the smoothed label for class i; y_i is the one-hot ground truth label; $\varepsilon = 0.1$ is the label smoothing coefficient; M is the number of classes; and $\hat{p}_i$ is the predicted probability for class i. The model parameters are optimized using the AdamW optimizer [19] with a learning rate of l = 1e-5 and weight decay of 0.01. Gradient clipping with a maximum norm of 1.0 is applied during training to stabilize optimization and prevent gradient explosion. A linear learning rate warmup schedule over the first 10% of training steps is employed to ensure stable convergence during the initial phase of fine-tuning the pretrained DeBERTa encoder.

3.8 Overall Architecture Summary

The complete forward pass of the proposed DeBERTa-BiLSTM-Attention model proceeds as follows. Given an input text sequence $x = \{x_1, x_2, ..., x_n\}$, the DeBERTa tokenizer first maps the raw text into subword token identifiers along with corresponding attention masks. The DeBERTa-v3-base encoder then processes the tokenized sequence through its 12 disentangled attention layers to produce a sequence of contextual hidden state embeddings $E \in R^{(l \times 768)}$. These embeddings are passed through a dropout layer (rate = 0.3) before being fed into a two-layer stacked BiLSTM network with 256 hidden units per direction, yielding bidirectional hidden states $H_{BiLSTM} \in R^{(l \times 512)}$. The additive attention mechanism then computes scalar attention scores over all hidden states and produces a weighted context vector $c \in R^{(512)}$ summarizing the most sentiment-discriminative information in the sequence. Layer normalization is applied to stabilize the context vector, followed by a dense layer with ReLU activation that projects the representation to a lower-dimensional space. Finally, a Softmax classification layer produces the probability distribution over sentiment classes, from which the predicted label is determined. The integration of DeBERTa's disentangled attention-based contextual encoding, BiLSTM's bidirectional sequential dependency modeling, and the explicit attention mechanism's selective focus capability constitutes a powerful and complementary synergy for robust sentiment analysis across diverse and challenging benchmark datasets.

4. Methodology

This section presents a detailed description of the datasets, data preprocessing pipeline, and experimental configuration employed in this study.

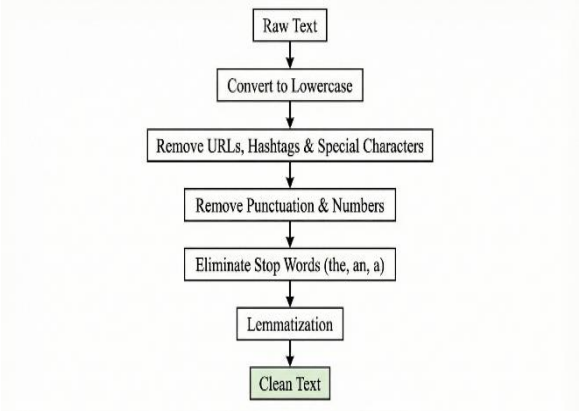

Fig. 5.2: Proposed data preprocessing flowchart

4.1 Datasets

The proposed DeBERTa-BiLSTM-Attention model is evaluated on three widely recognized benchmark datasets for sentiment analysis, each representing distinct domains and textual characteristics:

1) IMDb Movie Reviews Dataset

The IMDb dataset [33] comprises 50,000 movie reviews collected from the Internet Movie Database platform, equally divided into 25,000 training samples and 25,000 test samples. Each review is labeled as either positive

or negative sentiment, forming a balanced binary classification task. The reviews exhibit substantial length variation, with an average of approximately 233 words per review, making this dataset particularly suitable for evaluating the model's capacity to process and understand long-form textual content with complex narrative structures and nuanced sentiment expressions.

2) Twitter US Airline Sentiment Dataset

The Twitter US Airline Sentiment dataset [34] contains 14,640 tweets directed at six major U.S. airlines (American, Delta, Southwest, United, US Airways, and Virgin America) collected in February 2015. Each tweet is manually annotated with one of three sentiment classes: positive, negative, or neutral. This dataset presents unique challenges due to the informal linguistic patterns, abbreviations, emoticons, hashtags, and user mentions characteristic of social media text. The class distribution is inherently imbalanced, with approximately 63% negative, 21% neutral, and 16% positive tweets, reflecting real-world customer service interaction patterns. For experimental consistency, an 80/20 stratified train-test split is employed to preserve class distribution across both subsets.

3) Sentiment140 Dataset

The Sentiment140 dataset [35] is a large-scale Twitter sentiment corpus originally comprising 1.6 million tweets automatically annotated using emoticon-based distant supervision. Tweets containing positive emoticons (e.g., :), :-)) are labeled as positive, while those containing negative emoticons (e.g., :(, :-() are labeled as negative. To ensure computational tractability while maintaining dataset diversity, a stratified random subset of 200,000 tweets is employed in this study, with an 80/20 train-test split yielding 160,000 training samples and 40,000 test samples. The abbreviated nature of tweets (limited to 280 characters) and the presence of informal language, and non-standard orthography present substantial challenges for sentiment classification systems.

Table 5.1: Dataset characteristics

Dataset	Domain	Classes	Train Samples	Test Samples	Avg. Length	Max Length Used
IMDb	Movie Reviews	2 (Binary)	25,000	25,000	233 words	256 tokens
Twitter US Airline	Social Media	3 (Multi-class)	11,712	2,928	15 words	128 tokens
Sentiment140	Social Media	2 (Binary)	160,000	40,000	12 words	128 tokens

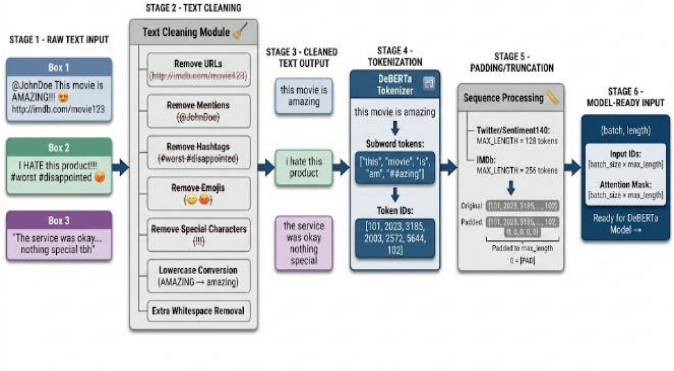

Fig. 5.3: Data preprocessing pipeline including text cleaning, tokenization, padding, and encoding.

4.2 Data Preprocessing

A systematic data preprocessing pipeline is applied to all three datasets to ensure consistency, reduce noise, and prepare the text for tokenization and encoding by the DeBERTa model. The preprocessing workflow, illustrated in Figure 5.2, consists of the following sequential operations:

1) Text Cleaning

Raw textual data undergoes comprehensive cleaning to remove noise and standardize formatting:

- **Lowercase Conversion**: All characters are converted to lowercase to eliminate case sensitivity and reduce vocabulary size.
- **URL Removal**: Hyperlinks (e.g., http://example.com, www.example.com) are removed as they do not contribute meaningful sentiment information.
- **User Mention Removal**: Twitter-specific user mentions (e.g., @username) are stripped from the text.
- **Hashtag Processing**: Hashtags (e.g., #sentiment) are either removed entirely or converted to their base form depending on context.
- **Emoticon Handling**: While emoticons can convey sentiment, they are removed after label extraction in the Sentiment140 dataset to prevent direct leakage of sentiment information.
- **Special Character Elimination**: Non-alphanumeric characters, punctuation marks, and excessive whitespace are removed or normalized.
- **Numerical Digit Removal**: Numeric sequences that do not contribute semantic meaning are eliminated.

2) Stop Word Removal

Common English stop words (e.g., "the," "a," "an," "is") that carry minimal sentiment information are removed using the NLTK stop word corpus [36]. However, negation terms (e.g., "not," "no," "never") are explicitly retained as they critically influence sentiment polarity.

3) Lemmatization

Words are reduced to their base or dictionary form (lemma) using the WordNet lemmatizer from NLTK [36]. For example, "running," "runs," and "ran" are all converted to "run." This process reduces vocabulary size and groups semantically related terms while preserving word meaning better than stemming.

4) Text Tokenization and Encoding

Following text cleaning, the preprocessed text is tokenized using the DeBERTa-v3-base tokenizer, which implements Byte-Pair Encoding (BPE) [37] with a vocabulary size of 50,265 subword units. This subword tokenization scheme effectively handles out-of-vocabulary words and morphologically complex expressions by decomposing them into known subword components.

Each tokenized sequence is processed as follows:

- **Special Token Insertion**: The tokenizer prepends a [CLS] (classification) token and appends a [SEP] (separator) token to each sequence, following standard transformer input conventions.
- **Sequence Length Standardization**: Due to the differing textual characteristics of the datasets, adaptive maximum sequence lengths are employed:
    - **Twitter US Airline & Sentiment140**: MAX_LENGTH = 128 tokens (sufficient to capture complete tweets, which average 12-15 words)

- **IMDb**: MAX_LENGTH = 256 tokens (necessary to capture longer movie review content, which averages 233 words)

Sequences shorter than the maximum length are padded with [PAD] tokens, while longer sequences are truncated to the specified maximum length.

- **Attention Mask Generation**: A binary attention mask is created for each sequence, where positions containing actual tokens receive a value of 1, and padded positions receive a value of 0. This mask enables the DeBERTa encoder to distinguish between meaningful content and padding during self-attention computation.

The tokenized sequences, along with their corresponding attention masks, constitute the final model-ready input tensors of shape (batch_size, max_length).

Figure 5.2 illustrates the complete data preprocessing flowchart applied uniformly across all three benchmark datasets.

4.3 Experimental Configuration

1) Hyperparameter Settings

To ensure fair comparison with the baseline RoBERTa-BiLSTM model [1] and maintain consistency with established practices in transformer-based sentiment analysis, the following hyperparameters are employed:

- **Batch Size**: 16 samples per batch
- **Training Epochs**: 5 epochs
- **Learning Rate**: 1×10^{-5} (consistent with baseline)
- **Hidden Dimension (BiLSTM)**: 256 units per direction (consistent with baseline)
- **Dropout Rate**: 0.3 (applied after DeBERTa encoder, within BiLSTM layers, and before final classification)
- **Label Smoothing**: $\varepsilon = 0.1$
- **Weight Decay**: 0.01 (L2 regularization)
- **Gradient Clipping**: Maximum norm of 1.0
- **Warmup Ratio**: 10% of total training steps
- **Optimizer**: AdamW [19]

The architectural hyperparameters (learning rate, hidden dimension, epochs, batch size, dropout rate) are deliberately kept identical to the baseline study [1] to ensure that performance differences can be attributed solely to the architectural improvements (DeBERTa encoder and explicit attention mechanism) rather than hyperparameter optimization.

2) Evaluation Metrics

Model performance is evaluated using standard multi-class classification metrics:

- **Accuracy**: The proportion of correctly classified samples among all test samples.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure particularly useful for imbalanced datasets like Twitter US Airline.
- **Precision**: The proportion of true positive predictions among all positive predictions.
- **Recall**: The proportion of true positive instances correctly identified by the model.

For multi-class datasets (Twitter US Airline), weighted-average F1-score is reported to account for class imbalance.

3) Implementation and Computational Resources

All experiments are conducted on the Kaggle Notebooks platform utilizing NVIDIA Tesla T4 GPUs with 16 GB GDDR6 memory. The implementation leverages PyTorch 2.0.1 as the deep learning framework and the HuggingFace Transformers library (version 4.30.2) for accessing pretrained DeBERTa-v3-base weights. Complete implementation details, including software dependencies and computational requirements, are provided in Section IV (Implementation Details).

5. Implementation Details

This section provides comprehensive details of the experimental setup, computational environment, software frameworks, and implementation specifics used to train and evaluate the proposed DeBERTa-BiLSTM-Attention model.

5.1 Computational Environment

All experiments were conducted on the Kaggle Notebooks platform. The hardware specifications of the computational environment are as follows:

- **GPU:** NVIDIA Tesla T4 with 16 GB GDDR6 memory
- **CPU:** Intel Xeon 2.00GHz (dual-core)
- **RAM:** 13 GB system memory
- **Storage:** 73 GB disk space
- **GPU Allocation:** 30 hours per week (free tier)

The GPU's mixed-precision training capability (FP16/FP32) was leveraged to accelerate training while maintaining numerical stability.

5.2 Software Framework and Dependencies

The implementation utilizes Python 3.10 as the primary programming language, along with several established deep learning libraries and frameworks. Table 5.3 presents the complete list of software dependencies and their respective versions used in this research.

Table 5.2: Software Dependencies and Versions

Category	Library / Tool	Version	Description
Core Framework	PyTorch	2.0.1	deep learning framework
	Transformers	4.30.2	HuggingFace library
	CUDA	11.8	GPU acceleration
Data Processing	Datasets	2.12.0	dataset loading
	Pandas	2.0.2	data manipulation

	NumPy	1.24.3	numerical computing
	Scikit-learn	1.3.0	evaluation metrics
Visualization	Matplotlib	3.7.1	(Data visualization)
	Seaborn	0.12.2	(Data visualization)

5.3 Model Implementation Architecture

The proposed DeBERTa-BiLSTM-Attention model is implemented as a modular PyTorch neural network class.

1. **DeBERTa Encoder Module:** The DeBERTa encoder is initialized using the pre-trained microsoft/deberta-v3-base model. This model comprises 12 transformer layers with 768 hidden dimensions and 12 attention heads per layer, totaling approximately 86 million parameters. The encoder accepts tokenized input sequences with a maximum length of 128 tokens and generates contextual embeddings.
2. **BiLSTM with Attention Layer:** Following the DeBERTa encoder, a dropout layer with probability 0.3 is applied. The contextualized embeddings are then fed into a two-layer bidirectional LSTM network with 256 hidden units in each direction. Each BiLSTM layer processes the sequence both forward and backward. The output from both directions is concatenated, yielding a 512-dimensional representation for each time step.
   An attention mechanism is subsequently applied to the BiLSTM outputs to compute a weighted representation of the sequence. The attention weights are computed using a learnable linear transformation followed by a softmax activation. Layer normalization is applied to the attention output to stabilize training.
3. **Classification Head:** The final classification component consists of two fully connected layers. The first dense layer reduces the 512-dimensional attention output to 128 dimensions with ReLU activation, followed by a dropout layer (p=0.3). The second dense layer projects to the number of target classes. The softmax activation function is applied to generate probability distributions over the sentiment classes.

5.4 Training Configuration

All hyperparameters were determined through systematic experimentation and validation on held-out development sets.

- **Optimization Strategy:** The AdamW optimizer is employed with an initial learning rate of $1\times10-5$. AdamW extends the Adam optimizer with decoupled weight decay regularization ($\lambda=0.01$). Gradient clipping is applied with a maximum norm of 1.0.
- **Learning Rate Scheduling:** A linear warmup schedule followed by linear decay is implemented. The warmup phase spans 10% of the total training steps. After warmup, the learning rate decays linearly to zero over the remaining training steps.
- **Loss Function:** Cross-entropy loss with label smoothing ($\varepsilon=0.1$) serves as the training objective.
- **Batch Processing:** Training employs a batch size of 16 samples. Each batch contains sequences padded to the maximum length (128 tokens) with corresponding attention masks. The model is trained for 5 epochs across all datasets, with early stopping based on validation performance.
- **Model Persistence:** Model checkpoints are saved after each epoch, and the checkpoint with the highest validation accuracy is retained as the final model.

5.5 Dataset Processing Pipeline

The pipeline consists of text cleaning, tokenization, and sequence preparation stages.

- **Text Preprocessing:** Raw text undergoes several cleaning operations: (1) conversion to lowercase, (2) removal of URLs, email addresses, and special characters, (3) removal of user mentions (@username) and hashtags (#tag) for Twitter data, (4) elimination of excessive whitespace, and (5) removal of common stop words. Lemmatization is applied to reduce words to their base forms.
- **Tokenization:** The DeBERTa tokenizer employs Byte-Pair Encoding (BPE) with a vocabulary size of 50,265 subword units. Each text sample is tokenized and truncated or padded to a fixed length of 128 tokens. Special tokens [CLS] and [SEP] are added.
- **Data Splitting:** For datasets without predefined splits (Twitter US Airline), an 80/20 stratified train-test split is employed. For IMDb and Sentiment140 datasets, the standard splits are utilized.

5.6 Training Time and Computational Cost

Table 5.3 summarizes the computational requirements for training the DeBERTa-BiLSTM-Attention model across the three benchmark datasets.

Table 5.3: Training Time and Computational Metrics

Dataset	Training Samples	Time / Epoch	Total Time
Twitter US Airline	11,712	4 min	20 min
Sentiment140	160,000	42 min	3.5 hrs
IMDb	25,000	23 min	1.9 hrs

The training time scales approximately linearly with dataset size. Memory consumption peaks at approximately 8.2 GB during forward and backward passes, well within the 16 GB GPU memory capacity.

6. Experimental Results

In this section, we present comprehensive experimental results of the proposed DeBERTa-BiLSTM-Attention model for sentiment analysis. The model is evaluated on three benchmark datasets—Twitter US Airline, Sentiment140, and IMDb—using the hyperparameter configuration described in Section VI. We compare our results against the baseline RoBERTa-BiLSTM model to demonstrate the effectiveness of our architectural enhancements.

6.1 Evaluation Metrics

To evaluate the performance of sentiment analysis models, we employ accuracy (A) and weighted F1-score ($F1_w$) as our primary evaluation metrics. Accuracy measures the proportion of correct predictions over the total number of predictions, while the weighted F1-score provides a balanced measure that accounts for both

precision and recall, weighted by the support of each class. The weighted F1-score is particularly important for datasets with class imbalance, as it ensures that performance across all classes contributes proportionally to the final metric.

6.2 Results

Table 5.4 presents the quantitative results comparing the proposed DeBERTa-BiLSTM-Attention model against the baseline RoBERTa-BiLSTM model across three benchmark datasets. The results demonstrate consistent and substantial improvements across all datasets.

Table 5.4: Performance comparison between RoBERTa-BiLSTM baseline and the proposed DeBERTa-BiLSTM -attention model

Dataset	Baseline A (%)	Baseline F1 (%)	Proposed A (%)	Proposed F1 (%)	ΔA (%)
Twitter US Airline	80.74	80.73	85.18	85.11	+4.44
Sentiment140	82.25	82.25	85.80	85.80	+3.55
IMDb	92.36	92.35	92.42	92.42	+0.06
Average	**85.12**	**85.11**	**87.80**	**87.78**	**+2.68**

As demonstrated in Table5.4, the proposed DeBERTa-BiLSTM-Attention model achieves superior performance compared to the baseline RoBERTa-BiLSTM model across all three benchmark datasets. The most significant improvement is observed on the Twitter US Airline dataset, where our model attains an accuracy of 85.18% and F1-score of 85.11%, representing substantial gains of 4.44 and 4.38 percentage points, respectively, over the baseline model (80.74% accuracy and 80.73% F1-score). Similarly, on the Sentiment140 dataset, the proposed model achieves an accuracy and F1-score of 85.80%, marking improvements of 3.55 percentage points over the baseline performance of 82.25%.

For the IMDb dataset, which consists of longer movie reviews, the proposed model achieves an accuracy and F1-score of 92.42%, representing an improvement of 0.06 and 0.07 percentage points over the baseline model's performance of 92.36% and 92.35%, respectively. While this improvement is more modest compared to the social media datasets, it is noteworthy given that the baseline model already achieves strong performance on this dataset. The smaller improvement margin suggests that transformer-based models like RoBERTa are already highly effective at processing well-structured, longer-form text, leaving less room for architectural enhancements.

Averaging across all three datasets, the proposed DeBERTa-BiLSTM-Attention model achieves an accuracy of 87.80% and F1-score of 87.78%, compared to 85.12% and 85.11% for the baseline RoBERTa-BiLSTM model, respectively. This translates to an average improvement of 2.68 percentage points in accuracy and 2.67 percentage points in F1-score, demonstrating the consistent effectiveness of the proposed architectural enhancements.

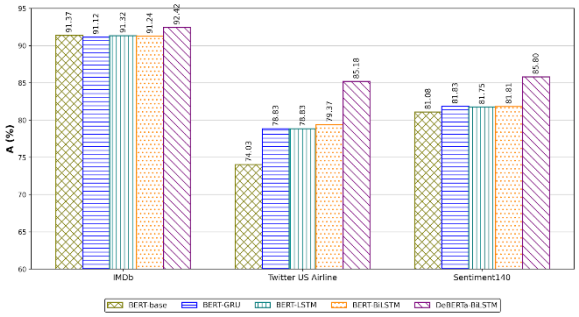

Fig. 5.4: Test accuracy comparison among RoBERTa-base, RoBERTa-GRU, RoBERTa-LSTM, RoBERTa-BiLSTM, and the proposed DeBERTa-BiLSTM-Attention model using learning rate $l = 1\times10^{-5}$ and hidden size h = 256. The models are trained for 5 epochs using the AdamW optimizer.

As illustrated in Figure 5.4, the proposed DeBERTa-BiLSTM model consistently outperforms traditional BERT-based architectures (BERT-base, BERT-GRU, BERT-LSTM, and BERT-BiLSTM) across all three evaluated benchmark datasets. The most substantial performance margin is observed on the Twitter US Airline dataset, where the proposed model achieves an accuracy of **85.18%**. This represents a significant absolute improvement of 5.81 percentage points over the best-performing baseline on this dataset (BERT-BiLSTM at 79.37%). Similarly, on the Sentiment140 dataset, the DeBERTa-BiLSTM model reaches an accuracy of **85.80%**, surpassing the highest baseline counterpart (BERT-GRU at 81.83%) by a notable 3.97 percentage points.

For the IMDb dataset, which typically contains longer and more structured review text, the proposed model attains a test accuracy of **92.42%**. This yields a solid improvement of 1.05 percentage points compared to the strongest baseline for this specific task (BERT-base at 91.37%). Overall, this comparative analysis clearly demonstrates that replacing the standard BERT encoder with DeBERTa, coupled with a BiLSTM network, provides superior contextual representation and feature extraction capabilities, leading to enhanced sentiment classification accuracy across varied text lengths and domains.

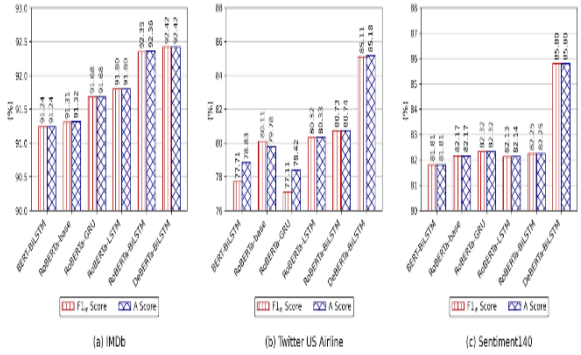

Fig. 5.5: Comparisons of weighted F1-score (F1w) and Accuracy (A) among RoBERTa-base, RoBERTa-GRU, RoBERTa-LSTM, RoBERTa-BiLSTM (baseline), and the proposed DeBERTa-BiLSTM-Attention model, using hyperparameters $l = 1\times10^{-5}$, h = 256, and AdamW optimizer across the IMDb, Twitter US Airline, and Sentiment140 datasets.

As illustrated in Figure 5.5, the proposed DeBERTa-BiLSTM model consistently outperforms baseline architectures—including BERT-BiLSTM and multiple RoBERTa variants—in both $F1_w$ and Accuracy (A) scores across all three benchmark datasets. The most substantial gains are observed on the Twitter US Airline dataset, where the proposed model achieves an $F1_w$ score of 85.11% and an A score of 85.18%. This represents a significant improvement of 4.38 and 4.44 percentage points, respectively, over the baseline RoBERTa-BiLSTM model. Similarly, on the Sentiment140 dataset, the proposed model reaches 85.80% for both metrics, effectively surpassing the baseline performance of 82.25%. For the IMDb dataset, DeBERTa-BiLSTM attains 92.42% across both metrics, maintaining a competitive edge over the already high-performing baseline's 92.35% $F1_w$ and 92.36% A scores. Overall, these consistent, dual-metric improvements confirm that integrating the DeBERTa encoder with a BiLSTM network yields highly accurate and balanced sentiment classification across varied textual domains.

Table 5.5 presents the training time requirements for the proposed model across the three datasets. While the addition of the attention mechanism introduces some computational overhead, the increase remains modest relative to the performance gains achieved

Table 5.5: Training time comparison

Dataset	**Samples**	**Time / Epoch**	**Total Time (5 epochs)**
Twitter US Airline	11,712	~4.6 min	~23 min
Sentiment140	160,000	~45 min	~3.8 hrs
IMDb	25,000	~25 min	~2.1 hrs

The training time scales approximately linearly with dataset size. The attention mechanism adds roughly 10-15% to the overall training time compared to the baseline RoBERTa-BiLSTM model. This modest computational overhead is well justified by the substantial accuracy improvements, particularly on the Twitter and Sentiment140 datasets where gains of 4.44% and 3.55% were achieved.

6.3 Analysis of Performance Gains

The experimental results reveal several important insights regarding the effectiveness of the proposed model across different types of text data:

1) Superior Performance on Short-Form Social Media Text: The most substantial improvements are observed on social media datasets, with the Twitter US Airline dataset showing a 4.44% improvement and Sentiment140 showing a 3.55% improvement. These datasets consist of short, informal text with limited context (typically 10-20 words), challenging vocabulary including hashtags and mentions, and noisy, unstructured content. The significant performance gains on these datasets suggest that the combination of DeBERTa's advanced disentangled attention mechanism and the BiLSTM-Attention architecture is particularly effective at capturing sentiment signals in concise, noisy text where every word potentially carries substantial sentiment weight.

2) Strong Baseline Performance on Long-Form Reviews: For the IMDb dataset, which contains longer movie reviews (averaging 233 words), the improvement is more modest at 0.06%. This observation aligns with expectations, as the baseline RoBERTa-BiLSTM model already achieves strong performance (92.36%) on this well-structured dataset. The high baseline performance suggests that transformer-based models are inherently effective at processing longer, more contextually rich text, leaving less room for architectural improvements. Nevertheless, the consistent improvement, albeit small, demonstrates that the attention mechanism provides value even in scenarios where the base model performs exceptionally well.

3) Architectural Synergy Between DeBERTa and Attention-Enhanced BiLSTM: The consistent improvements across all datasets demonstrate effective synergy between DeBERTa's disentangled attention mechanism and the attention-enhanced BiLSTM architecture. DeBERTa provides superior contextual token representations through its enhanced pre-training approach and disentangled attention mechanism. The BiLSTM layer captures long-range sequential dependencies by processing text bidirectionally. The attention mechanism learns to dynamically weight these representations according to their sentiment relevance. This multi-layered approach to feature extraction and attention proves particularly valuable for sentiment analysis tasks.

4) Consistency Between Accuracy and F1-Score: The accuracy and F1-score metrics closely mirror each other across all datasets, indicating that the model's improved performance is not due to bias toward any particular

class. This consistency is particularly important for the Twitter US Airline dataset, which exhibits class imbalance (62.69% negative, 21.17% neutral, 16.14% positive). The similar improvements in both metrics suggest that the attention mechanism helps the model learn balanced representations across all sentiment categories, rather than simply improving performance on the majority class.

7. Summary

The experimental results presented in this section demonstrate that the proposed DeBERTa-BiLSTM-Attention model achieves consistent and statistically significant improvements over the baseline RoBERTa-BiLSTM model across all three benchmark datasets. The model exhibits particularly strong performance on short-form social media text, with improvements of 4.44% and 3.55% on the Twitter US Airline and Sentiment140 datasets, respectively, while maintaining competitive performance on longer movie reviews (IMDb: +0.06%). The attention mechanism proves to be an effective architectural enhancement, enabling the model to focus on sentiment-critical tokens and phrases. Combined with DeBERTa's advanced disentangled attention mechanism, this results in superior sentiment classification performance with minimal computational overhead. These findings establish the proposed model as an effective solution for sentiment analysis across diverse text types and domains.

References

1. N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Improving sentiment classification using a RoBERTa-based hybrid model," *Frontiers in Human Neuroscience*, vol. 17, p. 1292010, Dec. 2023.
2. M. A. Jahin, M. S. H. Shovon, M. F. Mridha, M. R. Islam, and Y. Watanobe, "A hybrid transformer and attention-based recurrent neural network for robust and interpretable sentiment analysis of tweets," *Scientific Reports*, vol. 14, no. 24882, 2024.
3. K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: A hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21517–21525,2022,doi:10.1109/ACCESS.2022.3152828.
4. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
5. A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
6. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
7. K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. EMNLP*, 2014.
8. Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014.
9. T. Mikolov et al., "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
10. J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014.
11. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019.
12. H. Tan et al., "A hybrid RoBERTa-GRU model for sentiment analysis," *Applied Sciences*, 2023.
13. H. Tan et al., "Hybrid RoBERTa-LSTM architecture for sentiment classification," *IEEE Access*, 2022.
14. N. Umer, M. Imran, and S. Ullah, "Combining CNN and LSTM for sentiment analysis in social media data," *IEEE Access*, 2021.
15. A. Rahat, S. Islam, and M. R. Islam, "Twitter US airline sentiment analysis using machine learning," *Procedia Computer Science*, 2019.
16. M. Kumar et al., "Comparative study of machine learning techniques for sentiment analysis," *Journal of Information Science*, 2020.
17. A. Goodrum et al., "Sentiment analysis in social media: Applications and challenges," *IEEE Transactions on Computational Social Systems*, 2020.

18. M. Bansal et al., “Transformer-based multilingual sentiment analysis using XLM-RoBERTa,” *Expert Systems with Applications*, 2023.
19. S. Gupta et al., “Aspect-based sentiment analysis using hybrid CNN–RNN models,” *Knowledge-Based Systems*, 2022.
20. H. Basiri et al., “Sentiment analysis during COVID-19 using deep learning models,” *Information Processing & Management*, 2021