Sentiment Analysis with Deep Learning: Benchmarking CNNs vs. Prompt-Tuned BERT Models

Bo Huang and Fei Song*

School of Computer Science, University of Guelph, Guelph, Ontario, Canada
Email: bhuang06@uoguelph.ca (B.H.); fsong@uoguelph.ca (F.S.)
*Corresponding author

***Abstract*—With the growth of user-generated content on platforms like social networks and e-commerce, effective sentiment analysis has become increasingly important. This paper presents a comparative study of deep learning models for sentiment analysis, focusing on both Convolutional Neural Networks (CNNs) and prompt-based Bidirectional Encoder Representations from Transformers (BERT) models. We evaluated three CNN variants and multiple prompt strategies for BERT (hand-crafted, adaptive, and hybrid) across five diverse datasets. The results show that the hybrid Gated Recurrent Units_Attention Mechanism (GRU_ATT) BERT model outperforms all others. Prompt-based approaches, particularly adaptive and hybrid designs, offer strong performance and surpass CNN-Non-Static. Moreover, dataset characteristics, such as input length, class distribution, and language structure, significantly impact the performance of the models. Lastly, practical guidelines are proposed for selecting appropriate models based on dataset characteristics.**

***Keywords*—Convolutional Neural Network (CNN), Bidirectional Encoder Representations from Transformers (BERT), prompting, sentiment analysis**

I. Introduction

Sentiment analysis, also known as opinion mining, aims to extract and classify subjective information from text [1]. It plays a crucial role in understanding public opinion, user preferences, and emotional tone from sources such as product reviews, social media posts, and customer feedback. With the surge of online content in the form of user-generated reviews, tweets, and comments, accurate sentiment classification has become critical for applications in business intelligence, financial marketing, health care, and policy making [2].

Early approaches to sentiment analysis mainly relied on lexicon-based rules [3] and traditional machine learning models such as Naïve Bayes and Support Vector Machines (SVMs) [4]. While these models achieved moderate success, they depended heavily on handcrafted features and struggled to capture semantic and syntactic complexities inherent in natural language. Moreover, their performance varied significantly across domains and languages due to limited generalization capability.

With the advancement of deep learning, neural network models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated significant improvements in classification accuracy [5]. CNNs are effective in extracting local features and have been widely adopted for sentence-level sentiment tasks. More recently, transformer-based architectures, notably Bidirectional Encoder Representations from Transformers (BERT), have advanced Natural Language Processing (NLP) by enabling deeper contextual understanding through attention mechanisms [6].

This work conducts a comparative analysis of three CNN variants and multiple prompt-based BERT models, including hand-crafted, adaptive, and hybrid prompting techniques across five diverse datasets. The key contributions of this study are as follows:

(1) Benchmarking CNN Variants: We compare three CNN architectures—CNN-Random, CNN-Static, and CNN-Non-Static—and find that fine-tuning word embeddings (CNN-Non-Static) consistently improves performance across all datasets.

(2) Evaluation of Prompt-Tuned Transformers: We systematically assess seven BERT/RoBERTa-based models, including no-prompt baselines, hand-crafted prompts, adaptive prompting (LSTM/GRU_ATT), and hybrid strategies. Results show that transformer models, especially hybrid ones, significantly outperform CNNs in both accuracy and robustness across datasets.

(3) Statistical Confidence and Fairness Metrics: We report 95% confidence intervals for accuracy and emphasize macro F1-Scores to support statistically sound comparisons, particularly for imbalanced datasets such as Digital Music and Subscription Boxes.

(4) Practical Insights: We provide actionable model selection guidelines based on dataset characteristics, including class balance, sentence length, and linguistic complexity.

Manuscript received July 31, 2025; revised August 18, 2025; accepted September 15, 2025; published December 12, 2025.

doi: 10.12720/jait.16.12.1793-1808

II. Related Work

A. Sentiment Analysis Granularity

Sentiment analysis can be performed at varying levels of granularity, ranging from document-level to aspect-level, each offering different insights depending on the application context [4, 7].

Document-level sentiment analysis assigns an overall polarity (e.g., positive, negative, or neutral) to an entire document, such as a product review or article [8, 9]. While this provides a high-level summary, it lacks detail on specific opinions and is limited in scenarios with mixed or comparative sentiments [10].

Sentence-level analysis evaluates sentiment at the granularity of individual sentences, enabling finer distinctions when documents express varied opinions [11, 12]. This level is beneficial in detecting conditional or ambiguous statements and supports more localized interpretation than document-level analysis.

Phrase-level sentiment analysis targets specific expressions such as “great service” or “poor quality”, offering more nuanced sentiment detection [13, 14]. Techniques involve identifying polar expressions and resolving contextual polarity, which can achieve higher precision in interpretation.

Aspect-level sentiment analysis focuses on sentiments related to specific attributes or components (e.g., battery life, price) of a product or service [15]. It enables a detailed breakdown of user opinions and is increasingly adopted in practical applications requiring fine-grained feedback analysis.

Given the trade-off between granularity and complexity across sentiment analysis levels, this work focuses on the sentence level. It can offer a balance between detailed sentiment interpretation and practical feasibility. Sentence-level analysis enables more precise detection of polarity within individual opinions. This granularity aligns with the objectives of this study, which compares deep learning models in handling nuanced sentiment expressions.

B. Sentiment Analysis Methods

Fig. 1 presents an overview of the major methodologies employed in sentiment analysis [4, 16], which can be classified as lexicon-based approaches, traditional machine learning techniques, and state-of-the-art deep learning models [17].

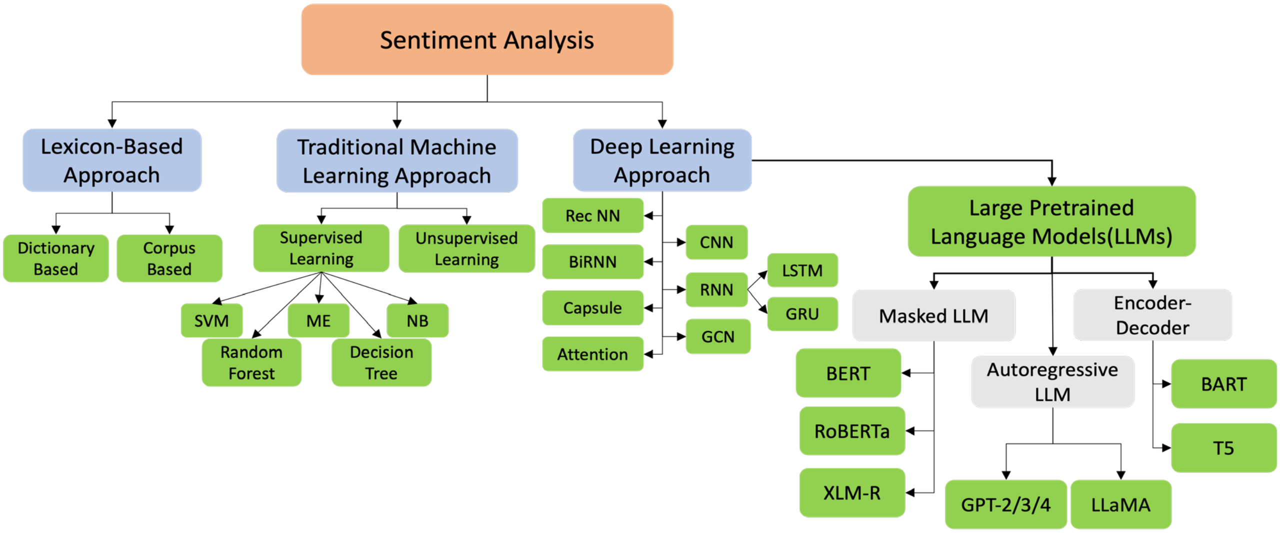


Fig. 1. Sentiment analysis methods.

Lexicon-based methods are based on pre-defined dictionaries of positive and negative words [18]. While efficient, they often fail to capture nuances and contextual dependencies. Traditional machine learning approaches, such as Naïve Bayes and Support Vector Machines (SVM), became popular by learning from labeled datasets [19, 20], but required manual feature engineering, such as selecting n-gram features, sentiment lexicons, or syntactic patterns.

Deep learning models removed the need for feature engineering. CNNs, introduced by Kim [21], became prominent due to their simplicity and effectiveness for sentence classification [5]. RNNs and LSTMs addressed long-term dependencies but were later surpassed by attention mechanisms and transformers [22].

BERT [6] introduced a bidirectional training approach, allowing models to capture contextual dependencies in text. Studies have demonstrated that fine-tuned BERT models outperform traditional deep learning architectures, such as CNNs and LSTMs, in sentiment classification tasks [23].

Prompt-based learning reframes classification as a language modeling task, allowing models to better leverage pre-trained knowledge. By designing templates or learning prompts, models can adapt to new tasks with little fine-tuning. Gao *et al.* [24] as well as Lester *et al.* [25] showed that prompting outperformed traditional fine-tuning.

Schick and Schütze [26] introduced Pattern-Exploiting Training (PET), a semi-supervised approach that integrates hand-crafted textual patterns with pre-trained language models such as RoBERTa. Their method achieved strong performance, reporting 89.6% accuracy on the AG's News dataset, and was also evaluated on additional datasets including Yelp, Yahoo, and MNLI. However, their work did not include an analysis of dataset characteristics or how such characteristics may influence model performance.

Varia *et al*. [27] proposed a multi-task prompting framework based on the T5 model, fine-tuned using instructional prompts. Their approach jointly tackled individual sub-tasks and the complete quadruple prediction task within a unified multi-task learning paradigm. Experiments on three benchmark datasets covering restaurant and laptop reviews demonstrated the generalizability of their method, which achieved a notable improvement of 8.29 F1 points in few-shot learning settings.

Memiş *et al.* [28] conducted a comparative analysis of deep learning models for financial sentiment analysis using Turkish tweets. They evaluated CNN, LSTM, GRU, and a GRU-CNN hybrid, focusing on how well each model captures sentiment in short, domain-specific texts. The GRU-CNN hybrid model yielded the highest classification performance, indicating the benefits of combining sequential memory with local convolutional features. However, their study is constrained to a single-language, single-domain context and does not explore generalizability to other languages or financial datasets.

Roumeliotis *et al.* [29] explored the use of Large Language Models (LLMs) such as GPT-4, BERT, and FinBERT for sentiment classification of cryptocurrency news articles. They fine-tuned these models using different optimizers (Adam and AdamW) and reported improvements particularly in neutral sentiment classification, a historically challenging class. Their results highlight the strength of LLMs in financial domains, though their work is limited by the use of proprietary APIs, a restricted fine-tuning dataset, and lack of longitudinal robustness testing.

While these studies contribute valuable insights into few-shot learning and prompt tuning for sentiment analysis, their scope is often narrow in terms of dataset diversity, metric consideration, or comparative benchmarking. Table I summarizes representative prior works, outlining models, datasets, contributions, and limitations.

To address these gaps, this study presents a comprehensive evaluation of CNN and prompt-based BERT models across a range of sentiment datasets with diverse characteristics. We systematically examine how factors such as text length, class imbalance, and linguistic complexity influence model performance. Based on these findings, we offer practical, data-driven guidelines for selecting appropriate models in real-world sentiment analysis tasks.

TABLE I. SUMMARY OF REPRESENTATIVE STUDIES ON SENTIMENT ANALYSIS USING DEEP LEARNING AND PROMPT-BASED METHODS

Reference	Year	Model	Datasets	Contributions	Limitation
Schick and Schütze [26]	2020	Pattern-Exploiting Training (PET) with RoBERTa	Yelp, AG's News, Yahoo, MNLI	Proposed PET for few-shot classification; demonstrated that PET RoBERTa significantly outperforms supervised and semi-supervised methods.	Performance depends on patterns; lacks robustness across domains and label distributions.
Varia *et al.* [27]	2022	Instruction-tuned T5 with ablation study	REST15, REST16, LAPTOP14	Introduced instruction tuning for few-shot aspect-based SA; reframed SA as text generation; achieved substantial performance gains (+8.29 F1) over prior methods	Limited to ABSA datasets; no broader cross-domain generalization; dataset characteristics underexplored
Memiş *et al.* [28]	2024	CNN, LSTM, GRU, GRU-CNN hybrid	Turkish financial tweets	Conducted comparative study of deep learning models for financial sentiment analysis; showed GRU-CNN variants achieved strong accuracy	Focused only on Turkish tweets; results not validated on multilingual or cross-domain financial texts
Roumeliotis *et al.* [29]	2024	GPT-4 (base and fine-tuned), BERT, FinBERT; fine-tuning with Adam/W optimizer	Crypto news datasets	Showed fine-tuning improves especially neutral-class detection	Used proprietary APIs (e.g., GPT-4); small sample size for fine-tuning; limited generalization due to single-domain focus.

III. METHODOLOGY

This section describes the model architectures, evaluation metrics and dataset preparation in our comparative analysis. This paper focuses on CNN variants and BERT-based models, each with unique mechanisms for classifying sentiment from text.

A. CNN Models

CNNs have been widely used for sentence classification due to their ability to extract n-gram features via convolutional filters. In this study, we adopt a CNN architecture composed of four layers: an input layer, a convolutional layer, a pooling layer, and a fully connected output layer. Fig. 2 presents the architecture adapted from Kim [21], commonly used for sentence-level classification.

The input layer represents each sentence as a word embedding matrix, where each row corresponds to a word transformed into a dense vector. We use pre-trained word2vec embeddings to encode semantic meaning into the input. These embeddings are either randomly initialized or loaded from fastText and optionally fine-tuned during training.

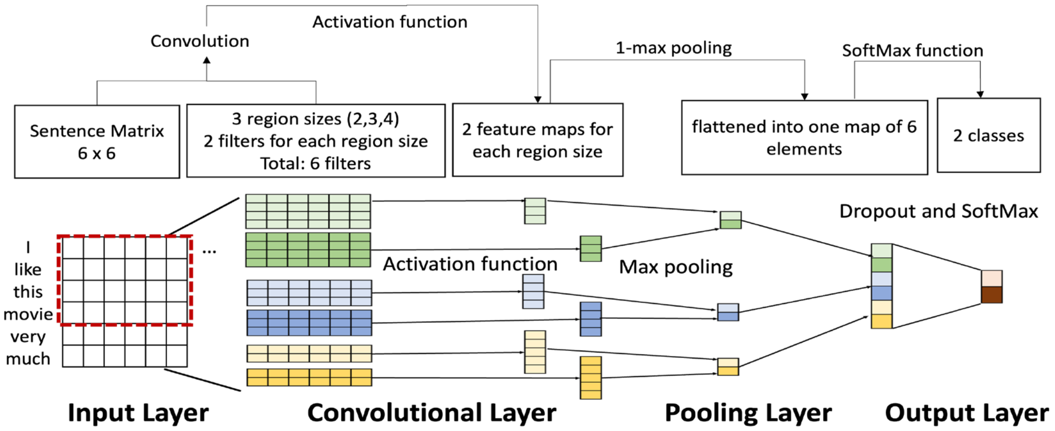


Fig. 2. The CNN Architecture used for sentence-level sentiment classification. The input sentence is transformed into an embedding matrix, followed by convolutional layers with multiple filter sizes to extract n-gram features. A max-pooling operation selects the most significant features, which are concatenated and passed through a fully connected layer. The final classification is performed using a SoftMax output layer.

The convolutional layer applies multiple filters of different sizes (e.g., 2, 3, and 4 words wide) to the word matrix to capture features at different granularities. Each filter slides over the sentence and performs element-wise multiplication followed by summation, thereby generating feature maps. Non-linearity is introduced via Rectified Linear Unit (ReLU) activation. For example, a 4×6 filter slides across the matrix and produces a 1×3 feature map, reflecting local dependencies in the text.

The pooling layer uses max-pooling to down-sample the feature maps by selecting the most significant features from each map. This step reduces dimensionality and computation while preserving important signals. Then, the resulting features are flattened and passed to a fully connected layer. This layer connects all extracted features to a set of output neurons, with weights being updated during training using backpropagation.

The output layer applies a SoftMax function to generate a probability distribution over sentiment classes (e.g., positive and negative). For instance, the sentence "I like this movie very much" may result in a vector [1, 0], indicating a positive sentiment. In our experiments, we evaluate three CNN variants: CNN-Random (randomly initialized embeddings), CNN-static (pretrained embeddings kept fixed), and CNN-non-static (pretrained embeddings updated during training).

- **CNN-Random**: Word vectors are initialized randomly and learned during training.
- **CNN-Static**: Use pre-trained word embeddings (e.g., fastText), which remain fixed during training.
- **CNN-Non-Static**: Use pre-trained embeddings, but allows them to be fine-tuned with task-specific data.

To ensure a fair and reproducible evaluation, we configured the CNN model using commonly adopted hyperparameter settings based on prior studies [5, 21]. The embedding dimension was set to 300, with convolutional filters of sizes 3, 4, and 5, each having 100 output channels. Dropout was applied with a rate of 0.5 to prevent overfitting. This study used the Adadelta optimizer with a learning rate of 0.01 and decay factor ρ = 0.95. These settings strike a balance between model expressiveness and training stability. A summary of all hyperparameters used for the CNN models is presented in Table II.

TABLE II. HYPERPARAMETERS FOR CNN SENTIMENT ANALYSIS

Hyperparameter	Value
Embedding Dimension	300
Filter Sizes	[3, 4, 5]
Number of Filters	[100, 100, 100]
Number of Classes	2
Dropout Rate	0.5
Optimizer	Adadelta
Learning Rate	0.01
ρ (Adadelta decay rate)	0.95

B. BERT Models

BERT is a transformer-based language model trained to capture bidirectional context. In this work, we employ two popular variants: BERT-base-uncased and RoBERTa-base.

To explore the effectiveness of prompt-based learning for sentiment analysis, we implement three prompt strategies inspired by the work of Zhang *et al.* [30]: handcrafted prompts, adaptive prompts, and hybrid prompts. A total of seven BERT-based model variants are evaluated, as illustrated in Fig. 3.

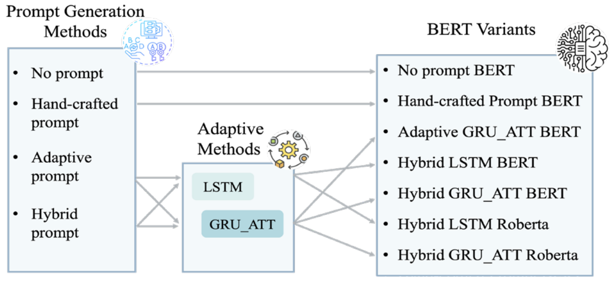


Fig. 3. Overview of the seven BERT variants evaluated in this study. The models include: (1) No Prompt BERT, (2) Hand-crafted Prompt BERT, (3) Adaptive GRU_ATT BERT, (4) Hybrid LSTM BERT, (5) Hybrid GRU_ATT BERT, (6) Hybrid LSTM RoBERTa, and (7) Hybrid GRU_ATT RoBERTa. These variants represent combinations of prompting strategies (none, hand-crafted, adaptive, hybrid) and backbone architectures (BERT-base-uncase or RoBERTa-base), enabling a comprehensive comparison across different configurations.

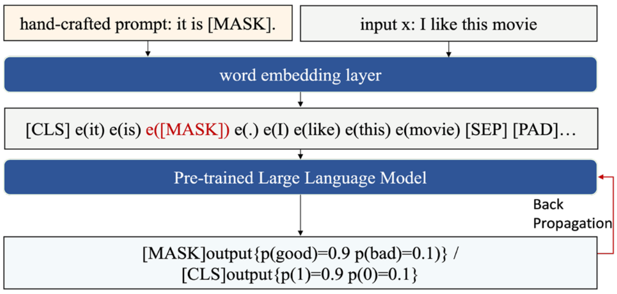

hand-crafted prompt: it is [MASK].
input x: I like this movie
word embedding layer
[CLS] e(it) e(is) e([MASK]) e(.) e(I) e(like) e(this) e(movie) [SEP] [PAD]…
Pre-trained Large Language Model
Back Propagation
[MASK]output{p(good)=0.9 p(bad)=0.1)} /
[CLS]output{p(1)=0.9 p(0)=0.1}

Fig. 4. Architecture of the Hand-crafted Prompt Model. A predefined textual template (e.g., "It is [MASK]") is prepended to the input sentence. The model then predicts the masked token or uses the [CLS] token for sentiment classification.

Hand-crafted Prompt: As shown in Fig. 4, a manually designed textual template is prepended to the input. For instance, a prompt such as "It is [MASK]" is constructed to steer the model's attention toward sentiment-related cues. The model is then trained to infer the masked token using contextual information from the entire input to predict whether the masked word is "good" or "bad". Two strategies are typically used for classification. The first strategy follows the standard BERT paradigm, utilizing the [CLS] token representation as an input to the classification layer. The second strategy exploits BERT's masked language modeling (MLM) objective, where the model predicts the masked token directly.

Adaptive Prompt: Unlike static templates, the adaptive prompt is generated dynamically by a learnable module, which takes the sentence embeddings as input and outputs a sequence of soft prompt tokens.

The process begins with the raw input text, which is tokenized and transformed into token embeddings via a standard embedding layer. These embeddings are then segmented into distinct components—"*CLS* + *Pattern* + *SEP* + *PAD*…"—as shown in Figs. 5 and 6. Specifically, the segment referred to as *[Pattern]* represents the embedding portion derived from the input tokens. From this segment, the corresponding *position_ids* are extracted and passed into a trainable adaptive module.

This adaptive module—implemented using either an LSTM or GRU_ATT network—generates a sequence of hidden states that act as learned soft prompt tokens. These adaptive hidden states are then inserted back into the embedding sequence at a designated location, replacing or augmenting the original prompt tokens. The resulting modified embeddings are subsequently passed into the BERT encoder.

From this point onward, the process mirrors that of the hand-crafted prompt model, with the BERT encoder producing contextualized representations that are used for sentiment classification. However, a key distinction lies in the training process. Backpropagation is applied not only to update the parameters of the BERT encoder but also to optimize the parameters within the adaptive prompt module. This joint training allows the adaptive component to dynamically adjust the prompts based on input semantics, thereby enhancing the model's ability to capture task-relevant contextual information.

We implement two variants of this module: one using LSTM networks (as shown in Fig. 5) and the other using GRU with attention mechanism (as shown in Fig. 6). These models, referred to as Adaptive LSTM BERT and Adaptive GRU_ATT BERT, are trained jointly with the BERT encoder and allow the prompts to adapt based on input semantics, offering greater flexibility in capturing nuanced sentiment patterns.

The two adaptive prompt methods are described as follows:

(1) Adaptive Prompt—LSTM

The Adaptive LSTM Prompt model is illustrated in Fig. 5. In this architecture, the prompt is dynamically generated using a learnable module built upon a LSTM network. This network is designed to capture sequential patterns and long-range dependencies in input text, allowing the model to adjust its internal prompt representations based on the semantic structure of the sentence.

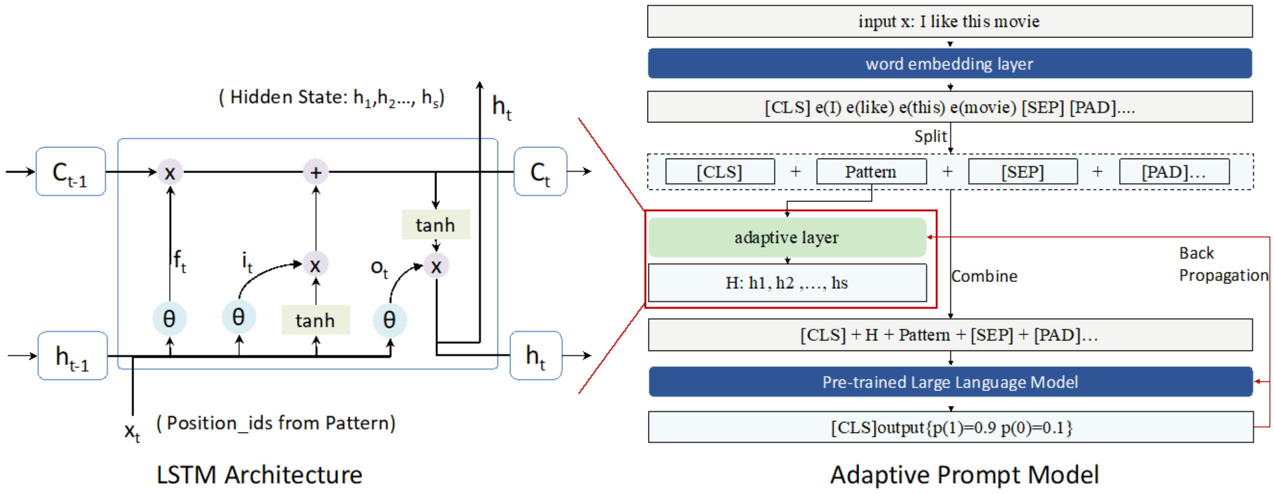

(Hidden State: h1,h2…, hs)
ht
Ct-1
Ct
x
+
tanh
ft
it
ot
x
x
θ
θ
tanh
θ
θ
ht-1
ht
xt
(Position_ids from Pattern)
LSTM Architecture
input x: I like this movie
word embedding layer
[CLS] e(I) e(like) e(this) e(movie) [SEP] [PAD]....
Split
[CLS]
+
Pattern
+
[SEP]
+
[PAD]…
adaptive layer
H: h1, h2 ,…., hs
Combine
Back Propagation
[CLS] + H + Pattern + [SEP] + [PAD]…
Pre-trained Large Language Model
[CLS]output{p(1)=0.9 p(0)=0.1}
Adaptive Prompt Model

Fig. 5. Architecture of the Adaptive LSTM BERT Model. The LSTM, as the adaptive layer, will generate a sequence of prompt embeddings (adaptive hidden states) from the input sentence, which are then prepended to the input embeddings before being passed into BERT. This allows dynamic generation of task-specific prompts that adjust based on input semantics.

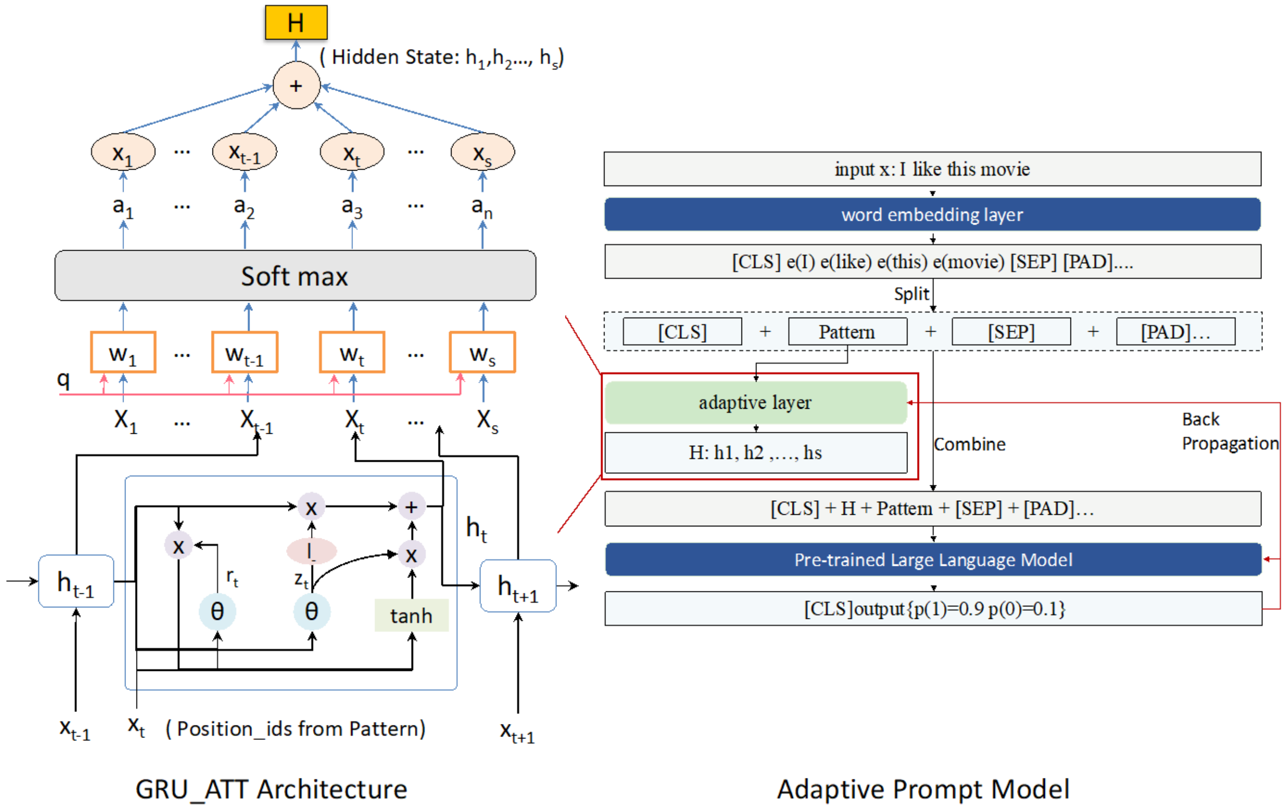

H
(Hidden State: h1,h2..., hs)
+
x1 ... xt-1 xt ... xs
a1 ... a2 a3 ... an
Soft max
w1 ... wt-1 wt ... ws
q
X1 ... Xt-1 Xt ... Xs
ht
ht-1
ht+1
x
rt
zt
1-
θ
tanh
xt-1
xt (Position_ids from Pattern)
xt+1
GRU_ATT Architecture
input x: I like this movie
word embedding layer
[CLS] e(I) e(like) e(this) e(movie) [SEP] [PAD]....
Split
[CLS] + Pattern + [SEP] + [PAD]...
adaptive layer
H: h1, h2 ,..., hs
Combine
Back Propagation
[CLS] + H + Pattern + [SEP] + [PAD]...
Pre-trained Large Language Model
[CLS]output{p(1)=0.9 p(0)=0.1}
Adaptive Prompt Model

Fig. 6. Architecture of the Adaptive GRU_ATT BERT Model. Similar to the Adaptive LSTM BERT in Fig. 5, this model uses a GRU-based encoder combined with an attention mechanism to generate dynamic prompt embeddings. The attention layer enhances the prompt generation by focusing on the most relevant parts of the input, allowing for improved context-aware sentiment classification.

The LSTM module operates by maintaining an internal memory cell C_t that evolves over time through the action of three gating mechanisms. Forget Gate (**f$_t$**) determines which parts of the previous memory state to discard, enabling the model to eliminate outdated or irrelevant information. Input Gate (**i$_t$**) controls how much of the new input is written into the memory cell, integrating relevant features from the current time step. Output Gate (**o$_t$**) regulates how much of the internal state is exposed as the current output, influencing the hidden state ht. These gates work collaboratively to produce adaptive hidden states that encode both contextual and positional information. These hidden states are subsequently injected as soft prompt tokens into the BERT encoder, where they modify the input representation in a learnable and data-driven manner.

By leveraging LSTM's ability to model sequential dependencies, the adaptive prompt mechanism enhances BERT's capacity to handle nuanced sentiment expressions, particularly in datasets with complex or longer text. This approach avoids the rigidity of hand-crafted templates and dynamically aligns prompt representations with the input semantics, improving flexibility and classification performance.

(2) Adaptive Prompt—GRU_ATT

The GRU_ATT Prompt model, shown in Fig. 6, augments the dynamic prompting mechanism by integrating a GRU with an attention layer. GRU simplifies the gating mechanism of LSTM by using two gates. Reset Gate (**r$_t$**) determines how much of the past information to forget at the current time step. Update Gate (**z$_t$**) balances the contribution of the previous hidden state and the candidate hidden state to form the new hidden representation.

Once the sequence of hidden states is computed, an attention mechanism is applied. The attention layer computes a set of importance weights that reflect the relevance of each token in the sequence. These weights are then used to produce a weighted sum of the hidden states, yielding a final context vector that selectively emphasizes informative parts of the input.

This architecture is particularly advantageous for sentiment classification, where certain tokens (e.g., negations, adjectives) carry more semantic weight. The attention-enhanced GRU enables the model to dynamically attend to such tokens, improving interpretability and predictive accuracy.

Compared to the LSTM-based adaptive model, GRU_ATT offers a more parameter-efficient design while maintaining high representational power. It performs particularly well on longer or multi-sentence reviews where important sentiment cues may appear in various parts of the input.

Hybrid Prompt: The hybrid strategy combines the strengths of both hand-crafted and adaptive approaches. A fixed template is first prepended to the input, followed by the dynamic hidden states generated by the adaptive module. This design ensures interpretability through human-readable templates while enhancing adaptability through learned prompt embeddings. The hybrid prompt workflow is visualized in Fig. 7.

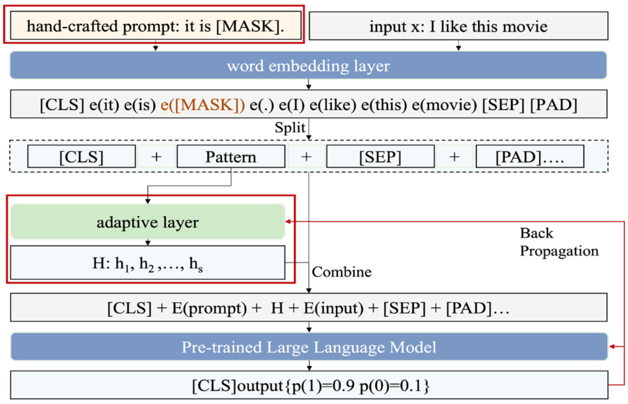

hand-crafted prompt: it is [MASK].
input x: I like this movie
word embedding layer
[CLS] e(it) e(is) e([MASK]) e(.) e(I) e(like) e(this) e(movie) [SEP] [PAD]
Split
[CLS] + Pattern + [SEP] + [PAD]….
adaptive layer
H: h1, h2,…, hs
Back Propagation
Combine
[CLS] + E(prompt) + H + E(input) + [SEP] + [PAD]…
Pre-trained Large Language Model
[CLS]output{p(1)=0.9 p(0)=0.1}

Fig. 7. Architecture of the Hybrid Prompt Model. It combines a fixed hand-crafted template with a learnable adaptive prompt generated via LSTM or GRU_ATT. The static and dynamic components are concatenated and injected into the BERT input embeddings.

Hyperparameters for BERT Variants: For all BERT-based models in this study, we adopted consistent hyperparameter settings to ensure fair comparisons. The learning rate was set to 5×10^{-6}, and training was performed using a batch size of 16. For hybrid models, two additional parameters, PATTERN and steps, were introduced to control how the auxiliary components interact with BERT's contextual embeddings.

The PATTERN parameter, set to 3, determines the number of prefix tokens in the input embedding sequence. It specifies where the output of the GRU_ATT or LSTM encoder is inserted within the final contextualized embedding. Specifically, the embedding is split at position PATTERN, and the sequential output is injected between the prefix and the remainder of the embedding sequence. This mechanism enables flexible positioning of auxiliary sequence modeling within the BERT framework.

The "steps" parameter, set to 4, is used in models with attention-based decoding layers (e.g., GRU_ATT). It controls the number of refinement iterations performed by the attention-based decoder. At each step, the decoder updates its internal state using GRU-based attention, thereby progressively refining the output representation. A higher "steps" value allows for deeper integration of contextual information but may increase computational cost.

These configurations were selected based on empirical tuning and align with the findings in [17], ensuring both model stability and performance across different datasets.

C. *Evaluation Metrics*

To assess model performance, this study adopts two primary evaluation metrics: Accuracy and Macro F1-Score. These metrics provide complementary insights into predictive effectiveness, especially when the datasets are imbalanced.

Accuracy measures the overall proportion of correctly classified instances among all predictions. It is a straightforward and widely used evaluation metric, formally defined as follows. TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (1)$$

While accuracy offers a general view of model performance, it may become unreliable when applied to imbalanced datasets, where the model could achieve high accuracy by predominantly predicting the majority class. To mitigate this limitation, the F1-Score is employed to provide a more balanced evaluation.

$$F1 = \frac{F1\_Positive + F1\_Negative}{2} \quad (2)$$

The *F1_Positive* score assesses the model's effectiveness in identifying positive samples and is computed as the harmonic mean of precision and recall:

$$F1\_Positive = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Precision = \frac{TP}{TP + FP}, \; Recall = \frac{TP}{TP + FN} \quad (4)$$

The *F1_Negative* score is introduced to evaluate how well the model identifies negative instances, which is critical for balanced binary classification. It is computed using the Negative Predictive Value (NPV) and the True Negative Rate (TNR):

$$F1\_Negative = 2 \times \frac{NPV \times TNR}{NPV \times TNR} \quad (5)$$

$$NPV = \frac{TN}{TN + FN}, \; TNR = \frac{TN}{TN + FP} \quad (6)$$

By incorporating both accuracy and macro F1 scores, this evaluation framework provides a more comprehensive assessment of model performance. It highlights not only overall correctness but also the model's sensitivity to both positive and negative sentiment classes—an essential aspect in tasks such as sentiment analysis, where class balance is polarized.

In addition to accuracy and macro F1-Score, we also report 95% Confidence Intervals (CIs) for accuracy values. A confidence interval provides an estimated range of values within which the true model performance (e.g., accuracy) is likely to fall, with a given level of confidence. Specifically, a 95% CI implies that if the same evaluation were repeated on many random samples, approximately 95% of the resulting intervals would contain the true accuracy.

For binary classification tasks, where accuracy is based on a proportion of correctly predicted instances over a fixed test set size *n*, we use the normal approximation to the binomial confidence interval, defined as

$$\text{CI} = \hat{p} \pm z \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad (7)$$

where $\hat{p}$ is the observed accuracy, n is the total number of test instances, and *z* is the *z*-score corresponding to the desired confidence level (1.96 for 95% confidence).

This formula assumes that the distribution of the accuracy estimate approximates a normal distribution, which is reasonable for sufficiently large sample sizes. Confidence intervals help quantify the reliability of reported metrics and offer a more statistically grounded comparison between models, especially when accuracy differences are small.

D. Datasets

Five binary-class datasets with varying characteristics were selected. Table III summarizes their properties.

TABLE III. DATASET STATISTICS AND SUBSET SIZES USED IN EXPERIMENTS

Dataset	Full Size	Subset Size	Pos (%)	Max L	Avg L	Sent. No.
SST2	9,602	9,602	51.6	56	19	1.00
Tweets	1.6 M	16,000	50.0	64	13	1.01
IMDB	50,000	10,000	50.0	2470	233	10.66
Sub. Boxes	16,000	16,216	64.1	1913	47.5	3.51
Music	124,000	12,404	92.5	4187	60	3.90

Note: Pos (%): proportion of positive samples; Max L: maximum input length; Avg L: average input length; Sent. No.: average number of sentences per sample.

To ensure fairness comparison, all models are evaluated using the same training/test splits and metrics. And all datasets were down sampled to similar sizes, as listed in Table III. This minimizes the effect of data volume and makes the evaluation focused on model architecture.

- **SST2**: It contains 9,602 movie review sentences. Each review is only one formal, structured sentence.
- **IMDB**: This dataset includes 50,000 movie reviews which are multi-sentence and relatively long (average 233 words).
- **Tweets (Sentiment140)**: Comprising 1.6 million tweets, this dataset reflects informal language typical of social media, including slang and incomplete sentences.
- **Amazon (Subscription Boxes and Digital Music**): These two datasets consist of Amazon product reviews collected in 2023. The class distribution in both datasets is imbalanced especially the proportion of positive reviews in Digital Music dataset is 92.51%.

E. Toolkits and Libraries Used

To ensure reproducibility and transparency, this study employed a suite of widely adopted NLP toolkits and deep learning libraries for the implementation and evaluation of CNN varriants and BERT-based sentiment analysis models. All experiments were conducted using Python 3.11.7, with CNN models developed in Jupyter Notebooks and BERT variants implemented through modular Python scripts. The core deep learning framework for both architectures was PyTorch, with HuggingFace Transformers used to load and fine-tune pretrained models such as bert-base-uncased and RoBERTa-base. CNN models utilized TorchText for data preprocessing and Gensim for integrating pretrained word embeddings (e.g., Word2Vec, fastText). Model evaluation was performed using Scikit-learn, and all result visualizations were generated using Matplotlib and Seaborn. A Conda-managed environment was used to ensure consistent package dependencies across experiments. Detailed hardware specifications are included to contextualize performance measurements. A full summary of the software libraries and their corresponding versions is provided in Appendix A (Table A1).

IV. RESULTS

This section presents the experimental results of CNNs and BERT variants evaluated on five benchmark sentiment analysis datasets: SST2, Sentiment140 (Tweets), IMDB, Amazon Subscription Boxes, and Amazon Digital Music. Both Accuracy and F1-Score are used as performance metrics.

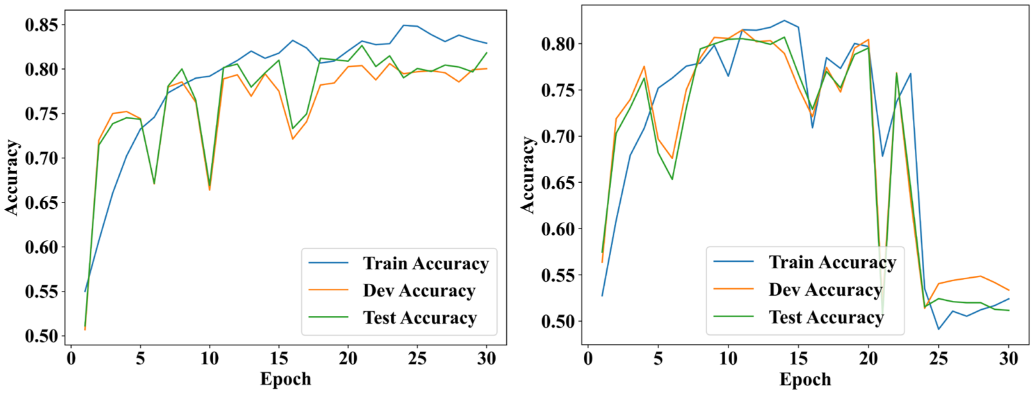
A. Hand-Crafted Prompt BERT

This section investigates the effect of different manually designed prompts on the performance of the BERT model for sentiment classification.

The choice of classification method affects model performance, as shown in Table IV and Fig. 8. Predicting the masked token can yield slightly better results for certain prompts—such as “The movie is [not] recommended” and “It makes me feel [good/bad]”.

TABLE IV. MODEL PERFORMANCE ACROSS DIFFERENT CLASSIFICATION

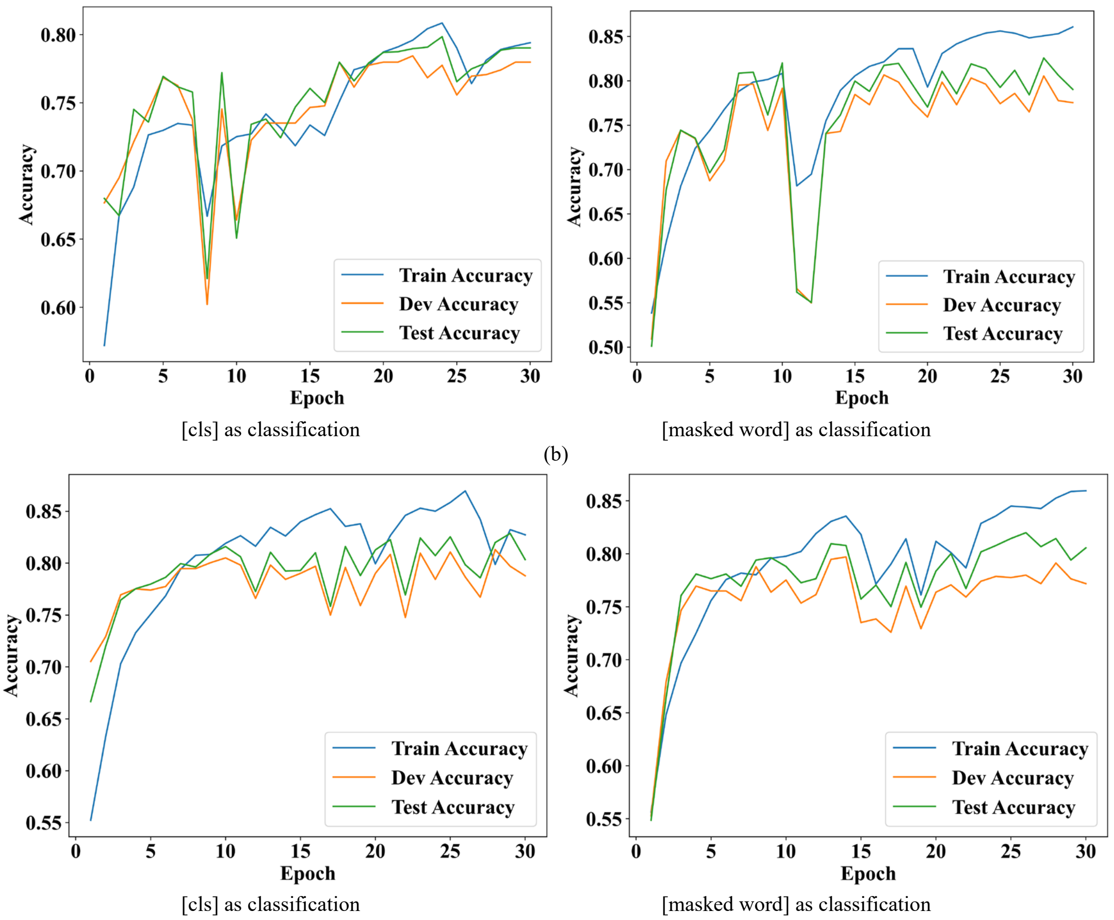
Prompts	[CLS] Token	Masked Word
The movie is [not] recommended	80.62%	81.48%
It makes me feel [good/bad]	78.44%	80.56%
It is [good/bad]	81.31%	78.47%



[cls] as classification

[masked word] as classification

(a)



[cls] as classification

[masked word] as classification

(b)

[cls] as classification

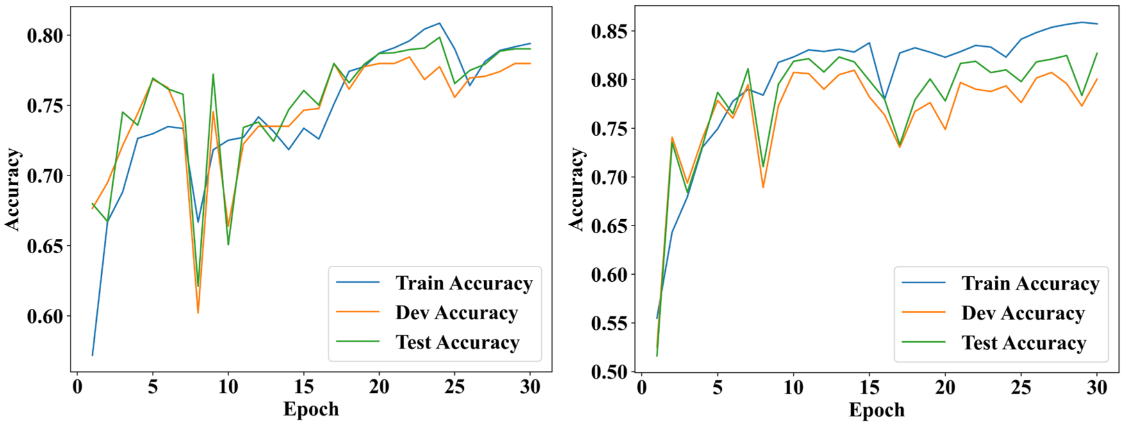
[masked word] as classification

(c)

Fig. 8. Performance comparison of Hand-crafted Prompt BERT models using two classification strategies across three prompt templates: (a) "The movie is [not] recommended", (b) "It makes me feel [good/bad]", and (c) "It is [good/bad]". For each prompt, the left subfigure shows results using the [CLS] token for classification, while the right subfigure shows results using [MASK] token prediction. This comparison illustrates how prompt design and classification method jointly affect model performance.

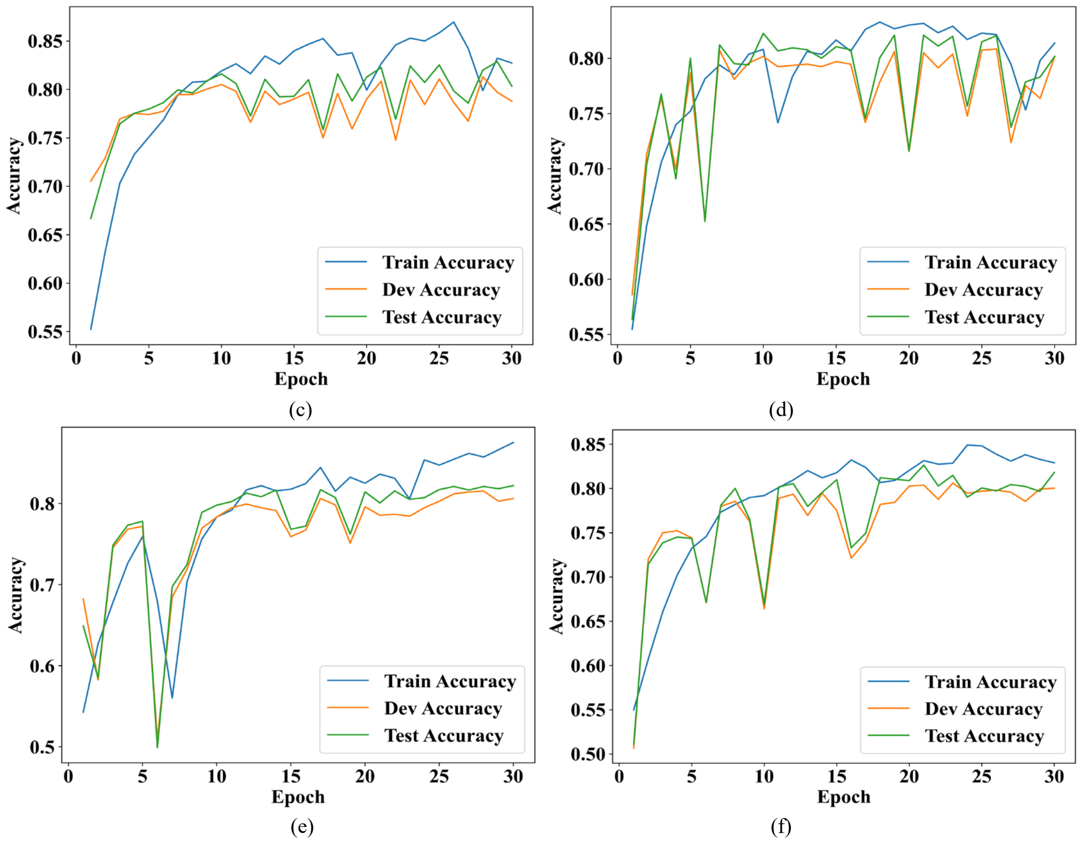
However, as shown in Fig. 8(a), masked-token prediction exhibits instability during training and can lead to convergence at suboptimal performance. The accuracy fluctuates more significantly and shows a tendency to get trapped in local optimum. For consistency and robustness, the [CLS]-based classification was employed for all subsequent experiments.

As illustrated in Fig. 9 and Table V, different prompt templates influence the results on the SST2 dataset. Among the six prompts tested, the prompt "The sentiment of the text is [positive/negative]" achieved the highest accuracy at 81.54%, and the performance may improve further with extended training. While the prompt "It makes me feel [good/bad]" produced the lowest accuracy of 78.44%, indicating a performance gap of 3.1% compared to the best performing prompt



(a)

(b)



(c)

(d)

(e)

(f)

Fig. 9. Performance of hand-crafted prompt BERT models using different prompt templates on SST2. (a) It makes me feel [good/bad]; (b) It was [good/bad]; (c) It is [good/bad]; (d) The sentiment is [positive/negative]; (e) The sentiment of the text is [positive/negative]; (f) The movie is [not] recommended. Each subplot illustrates the accuracy trend during training for a specific hand-crafted prompt, highlighting how template phrasing affects model performance.

TABLE V. ACCURACY OF HAND-CRAFTED PROMPT BERT MODEL WITH DIFFERENT PROMPTS ON SST2

Prompts	Accuracy
it makes me feel [good/bad]	78.44%
it was [good/bad]	80.96%
it is [good/bad]	81.31%
the sentiment is [positive/negative]	80.85%
the sentiment of the text is [positive/negative]	81.54%
the movie is [not] recommended	80.62%

B. *Comparison of CNN and Bert-Base-Uncased Models*

In this subsection, we evaluated eight models in total: three CNN variants (Random, Static, and Non-Static) and five BERT-base-uncased models with different prompting strategies—No Prompt, Hand-crafted, Adaptive GRU_ATT, Hybrid LSTM, and Hybrid GRU_ATT. The performance results are summarized in Tables VI and VII.

TABLE VI. ACCURACY COMPARISON ACROSS MODELS AND DATASETS

Model		SST2	Tweets	IMDB	Sub.Boxes	Music
CNN Models	CNN-Random	73.63%	75.34%	87.80%	90.52%	93.73%
	CNN-Static	85.98%	78.21%	90.70%	92.73%	95.30%
	CNN-NonStatic	86.48%	78.84%	90.30%	92.94%	95.56%
BERT-base Models	No Prompt	78.87%	72.95%	78.43%	87.41%	92.26%
	Hand-crafted	80.25%	73.15%	81.13%	88.43%	92.03%
	Adaptive GRU_ATT	90.77%	80.20%	90.55%	93.06%	95.04%
	Hybrid LSTM	91.59%	80.41%	91.19%	93.06%	96.04%
	Hybrid GRU_ATT	92.20%	81.50%	91.35%	93.29%	96.27%

TABLE VII. F1 SCORE COMPARISON ACROSS MODELS AND DATASETS

Model		SST2	Tweets	IMDB	Sub.Boxes	Music
CNN Models	CNN-Random	73.43%	75.32%	87.73%	87.76%	74.32%
	CNN-Static	85.79%	78.21%	90.63%	91.16%	83.56%
	CNN-NonStatic	86.35%	78.83%	90.25%	91.32%	84.71%
BERT-base Models	No Prompt	78.87%	72.84%	78.26%	83.31%	81.89%
	Hand-crafted	80.25%	72.74%	80.86%	91.84%	85.98%
	Adaptive GRU_ATT	90.76%	80.18%	91.04%	92.09%	86.76%
	Hybrid LSTM	91.59%	80.42%	90.87%	91.45%	87.07%
	Hybrid GRU_ATT	92.20%	81.50%	90.67%	91.63%	87.48%

Among the CNN models, CNN-Non-Static performed the best overall, except on the IMDB dataset, where CNN-Static slightly outperformed the former. This suggests that fine-tuning word embeddings helps more on shorter or informal texts, such as Tweets and SST2, where task-specific adaptation is beneficial. In contrast, for longer and more semantically rich inputs like IMDB reviews, the frozen pre-trained embeddings used in CNN-Static (e.g., fastText) already encode robust linguistic features that are well-suited for sentiment classification, resulting in competitive or even superior performance. The minimal deviation between CNN-Static and CNN-Non-Static on IMDB also suggests that embedding stability may be more important than adaptability for long-form texts. This trend is consistent across other datasets such as Subscription Boxes and Digital Music, where CNN-Non-Static retains a slight edge, but the performance gap is narrower—indicating that the benefit of embedding fine-tuning may diminish when pre-trained vectors already align closely with the domain-specific language.

Building on the CNN results, we take CNN-Non-Static—the strongest CNN variant—as a comparative baseline to assess the effectiveness of BERT variants. As illustrated in Fig. 10, we find that both Adaptive and Hybrid BERT models consistently achieve higher accuracy across all five datasets. In particular, Hybrid GRU_ATT yields the best accuracy on every dataset, indicating the advantage of combining static prompts with learned components. From Table VII, it shows that these improvements extend to F1-Scores as well: Hybrid GRU_ATT leads on SST2, Tweets, and Digital Music, while Adaptive GRU_ATT achieves the highest F1 on IMDB and Subscription Boxes.

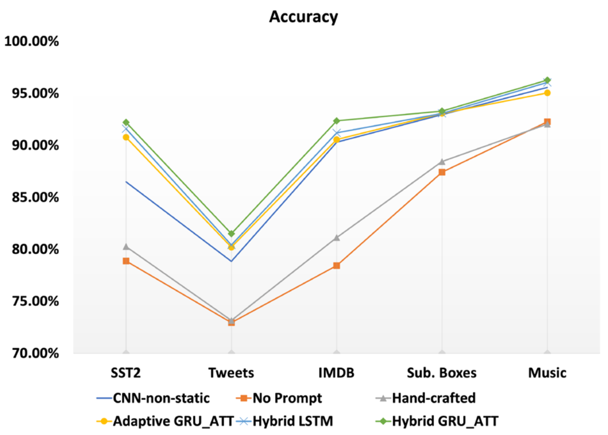


Fig. 10. The accuracy of BERT variants compared to the CNN-non-static model across all datasets.

This distinction appears to be linked to dataset characteristics. The IMDB and Subscription Box datasets contain longer input sequences, which provide richer semantic context for the adaptive prompt module to leverage, thereby reducing the relative benefit of hand-crafted prompt templates. Conversely, shorter or noisier texts (e.g., Tweets and SST2) benefit from the structural guidance offered by static templates—making hybrid approaches particularly effective in those cases.

A deeper comparison of BERT variants reveals further insights. Starting with the baseline BERT model without prompts, we observe that adding hand-crafted prompts provides consistent, albeit modest, performance gains in both accuracy and F1-Score across four datasets. The largest gain is observed on IMDB (+2.7% accuracy), while Tweets see a marginal improvement (+0.2%). However, on the Digital Music dataset, the hand-crafted prompt slightly underperforms (−0.23%), suggesting that rigid templates may not generalize well to domain-specific language distributions.

The Adaptive GRU_ATT BERT model, which generates prompts dynamically based on input semantics, significantly outperforms both no-prompt and hand-crafted prompt configurations. For instance, it improves IMDB accuracy from 78.43% to 90.55%, and SST2 from 78.87% to 90.77%. Even on strong baselines like Digital Music, it provides further boosts in accuracy (+2.78%). These results underscore the importance of prompt flexibility and input-aware design in enhancing model interpretability and performance.

Finally, the Hybrid prompting strategy—which combines static templates with adaptive modules—emerges as the most robust across all scenarios. The Hybrid GRU_ATT model achieves the best accuracy and F1-Scores on nearly every dataset, benefiting from both structured priors and dynamic adaptation. This approach offers a compelling solution for real-world sentiment classification tasks, as it balances manual prompt reliability with automated learning capacity, resulting in strong generalizability across domains and text types.

C. *Confidence Interval Analysis*

To assess the statistical reliability of model performance, we computed 95% CIs for accuracy across all model–dataset combinations. These intervals represent the range within which the true model accuracy is expected to fall with 95% confidence.

As shown in Appendix A (Table A2), Figs. 11 and 12, hybrid BERT models consistently achieved the narrowest confidence intervals, indicating stable performance. For instance, the Hybrid GRU_ATT model on the SST2 dataset achieved 92.20% accuracy with a CI of [90.97%, 93.43%], suggesting high statistical confidence in its superiority. Similarly, on the IMDB dataset, the same model yielded 91.35% accuracy with a CI of [90.25%, 92.45%].

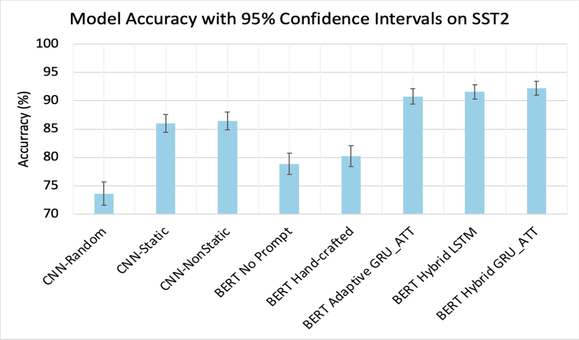


Fig. 11. Model accuracy with 95% confidence intervals on SST2.

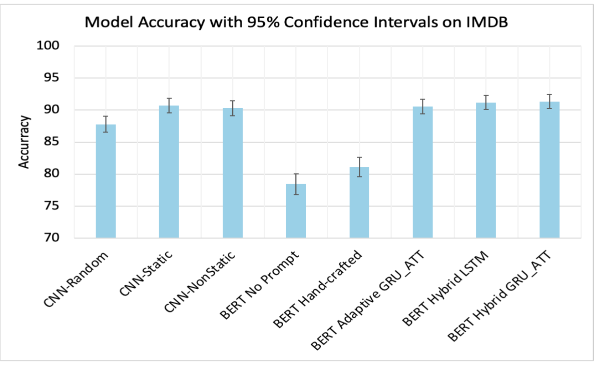


Fig. 12. Model accuracy with 95% confidence intervals on IMDB.

In contrast, models with lower accuracies or smaller performance margins—such as CNN-Random or No Prompt BERT—exhibited wider intervals. For example, CNN-Random on SST2 had an accuracy of 73.63% with a CI of [71.61%, 75.65%], reflecting more uncertainty and overlap with other models.

Moreover, in datasets with longer texts (e.g., IMDB, Sub.Boxes), confidence intervals tend to be tighter across all models, suggesting more consistent predictions due to richer input signals. On the other hand, datasets like Tweets, which are shorter and noisier, produced broader intervals, especially for simpler models like CNN-Random or BERT without prompting.

Overall, the inclusion of confidence intervals confirms the robustness of observed model rankings. In most cases, the Hybrid BERT variants (especially GRU_ATT) not only achieved the highest accuracy but also demonstrated statistically significant performance margins over baselines, as reflected by non-overlapping CIs with weaker models.

D. Impact of Dataset Size

As shown in Table VIII, CNN-non-static significantly benefits from larger training datasets. When trained on the full dataset, its performance surpasses that of the Hybrid GRU_ATT model trained on subsets. This suggests that CNN architectures rely more heavily on large datasets to learn effective representations, while prompt-based BERT models demonstrate greater sample efficiency.

TABLE VIII. PERFORMANCE OF CNN-NON-STATIC (FULL/SUBSET) VS. HYBRID GRU_ATT (SUBSET)

Dataset	CNN-Full		CNN-Subset		Hybrid-Subset	
	Acc.	F1	Acc.	F1	Acc.	F1
Tweets	84.10%	84.10%	78.84%	78.83%	81.50%	81.50%
Music	96.93%	87.86%	95.56%	84.71%	96.27%	87.48%
IMDB	92.55%	92.56%	90.30%	90.25%	91.35%	90.67%
SST2	86.48%	86.35%	–	–	92.20%	92.20%
Sub. Boxes	92.94%	91.32%	–	–	93.29%	91.63%

To further investigate the scalability of the Hybrid GRU_ATT model, we varied the proportion of the SST2 dataset used during training. As shown in Fig. 13, model accuracy increases steadily with more training data, peaking at 80% of the dataset. Interestingly, when using the full dataset (100%), a slight drop in accuracy is observed, possibly due to overfitting or noise introduced by marginal examples, indicating diminishing returns beyond a certain data threshold. Interestingly, training time also rises substantially with dataset size, reflecting greater computational demands. Since all experiments were conducted on the same hardware platform, training time can also serve as a proxy for energy consumption. From this perspective, using 60%–80% of the dataset not only achieves near-peak accuracy but also offers better energy efficiency, making it a more practical choice for deployment in resource-constrained or sustainability-aware environments.

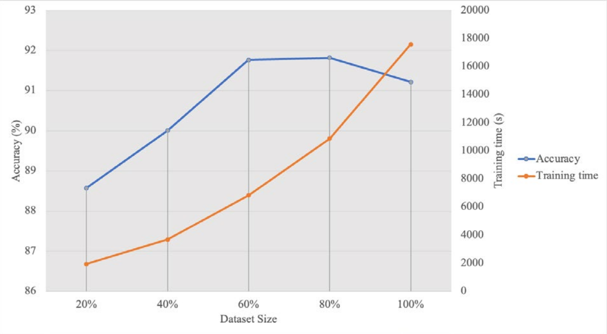


Fig. 13. Accuracy and training time for Hybrid GRU_ATT BERT-base model on different SST2 subset sizes.

E. Impact of Model Size

Regarding the Hybrid Prompt models, we also explore roberta-base as a pretrained model. The results are presented in Tables IX and X. The hybrid GRU_ATT BERT model consistently outperforms the hybrid LSTM BERT model across all datasets in bert-base-uncased model, while the roberta-base model shows a different trend. Specifically, the hybrid LSTM. Roberta model performs slightly better than the hybrid GRU_ATT Roberta model on the Tweets and Digital Music datasets.

TABLE IX. ACCURACY COMPARISON OF BERT AND ROBERTA WITH HYBRID PROMPTING (LSTM AND GRU_ATT)

Dataset	H(LSTM)		H(GRU_ATT)	
	BERT	RoBERTa	BERT	RoBERTa
SST-2	91.59%	93.08%	92.20%	94.23%
Tweets	80.37%	81.84%	81.50%	81.46%
IMDB	91.19%	93.40%	91.35%	93.88%
Sub. Boxes	93.06%	93.95%	93.29%	94.04%
Music	96.04%	96.69%	96.27%	96.52%

TABLE X. F1 COMPARISON OF BERT AND ROBERTA WITH HYBRID PROMPTING (LSTM AND GRU_ATT)

Dataset	H(LSTM)		H(GRU_ATT)	
	BERT	RoBERTa	BERT	RoBERTa
SST2	91.59%	94.23%	92.20%	93.07%
Tweets	80.42%	81.46%	81.50%	81.83%
IMDB	90.87%	93.88%	90.67%	93.40%
Sub. Boxes	91.45%	92.63%	91.63%	92.64%
Music	87.07%	88.20%	87.48%	88.89%

From Table IX, we also can observe that RoBERTa consistently performs better bert-base-uncased across all datasets when using an additional LSTM layer. For the GRU_ATT architecture, RoBERTa achieves better results in most cases, except for the Tweets dataset, where performance is nearly equal.

F. Impact of Dataset Characteristics

A comparative analysis from Tables III, VI, and VII highlights how specific dataset characteristics affect model performance. These effects are further summarized in Tables XI and XII.

TABLE XI. DATASET CHARACTERISTICS IMPACT

Observation	Impact
SST2, a relatively short and balanced dataset with single-sentence inputs, also shows strong gains from adaptive and hybrid prompts.	Simpler input structure makes it easier for prompts to be effective, with the hybrid model achieving the highest accuracy (92.20%).
Tweets is informal language with relatively low average sentence length (13) and balanced sentiment (50%).	Despite its size, performance improvements are modest, suggesting that large datasets with short, noisy, or informal text (like tweets) benefit less from hand-crafted prompts and more from adaptive or hybrid methods. CNN-non-static training with full dataset can show the best performance.
IMDB has long documents (avg. 233 words) and is balanced.	Performance greatly improves with adaptive prompts (from ~78% to over 91%). This suggests that long texts provide richer context, allowing dynamic prompt models to better capture sentiment nuances.
Subscription Boxes is multi-sentence, slightly imbalanced (64% positive), and mid-sized (16K samples).	Good performance across all models, and hybrid methods edge out others slightly, showing that both structure and adaptation help in moderately sized review-style datasets.
Digital Music has the highest positive label ratio (92.51%), and longer average length (60).	All models perform exceptionally well on this dataset. The class imbalance might make accuracy less informative, but high F1-Scores (~96%) indicate strong precision-recall performance. Hybrid prompts show the best results.

TABLE XII. PROMPT SENSITIVITY AND DATASET TYPE

Dataset	Prompt Sensitivity	Insights
SST2	High	Single-sentence dataset with clear sentiment benefits from prompt design, particularly the hybrid setup.
Tweets	Moderate	Short, noisy text benefits moderately. Adaptive prompts are better than hand-crafted, suggesting context modeling is crucial.
IMDB	High	Adaptive and hybrid prompts drastically improve performance, showing the benefit of dynamic contextual understanding for long reviews.
Subscription Boxes	Low-Moderate	Already high baseline performance; hybrid prompts still offer small improvements.
Digital Music	Low	High baseline due to skewed class distribution; prompts offer diminishing returns, but hybrid models still slightly boost results.

In conclusion, three key dataset characteristic influence model performance:

- **Input Length**: Datasets with longer average input lengths, such as IMDB (average length: 233), enable adaptive and hybrid prompt-based models to utilize more contextual information. This results in over 12% accuracy improvements over the no-prompt BERT model. It demonstrates the advantage of prompt-aware models in handling rich textual inputs.
- **Class Distribution**: In datasets with significant class imbalance, such as Digital Music, models often report high accuracy (up to 96.27%); however, the F1-Score (87.48%) offers a more nuanced view by capturing the trade-off between precision and recall, making it a more reliable performance metric in such scenarios.
- **Textual Structure and Formality**: Structured and formal datasets, like Subscription Boxes reviews, benefit substantially from prompt-based methods. In contrast, datasets characterized by informal and noisy language—such as Tweets—tend to get lower performance across all models, with accuracy generally around 80%, highlighting the challenges of sentiment analysis in less structured domains.

V. GUIDELINES

This section presents practical guidance based on our comparative analysis of ten models. These include three CNN variants (Random, Static, Non-Static) and seven BERT-based configurations: No Prompt, Hand-crafted Prompt, Adaptive GRU_ATT, Hybrid LSTM, Hybrid GRU_ATT, Hybrid LSTM RoBERTa, and Hybrid GRU_ATT RoBERTa.

A. Model and Prompting Strategy Selection

The strengths and trade-offs of each model architecture and prompting method are summarized as follows. These insights aim to guide model selection based on resource availability and task complexity.

- **CNN-Non-static** offers speed and efficiency, making it suitable for lightweight tasks. However, it falls short in capturing complex semantics, especially in longer or informal texts.
- **BERT without prompting** provides a solid baseline but lacks task-specific tuning. Its performance is generally stable but rarely competitive with more tailored approaches.
- **Hand-crafted prompts** yield modest gains (typically 1–3%) and are easy to implement, especially when domain expertise is available.
- **Adaptive prompting** using GRU_ATT significantly boosts performance, particularly on datasets with long or nuanced inputs. On IMDB, for instance, it improves accuracy by over 12% compared to the BERT baseline.
- **Hybrid prompting** which blends structured templates with adaptive learning, consistently delivers the strongest results. Hybrid GRU_ATT stands out in both accuracy and F1 across all datasets tested.

B. Dataset-Specific Insights

Each dataset presents distinct linguistic and structural characteristics. The following recommendations reflect how different models adapt to these traits in real-world text.

- **Tweets**: Informal and noisy text benefits most from adaptive or hybrid prompts. Hybrid GRU_ATT performs the best (81.50%), although CNN-non-static offers a quick and competitive alternative.
- **IMDB**: Reviews are lengthy and balanced. CNNs struggle with long-range context, while adaptive and hybrid prompting show clear advantages.
- **SST2**: The short and well-structured nature of this dataset favors BERT-based models. Hybrid configurations yield the best performance (92.20% accuracy and F1).
- **Subscription Boxes**: Sentiment is more subtle. Adaptive GRU_ATT performs consistently well in capturing nuanced patterns.
- **Digital Music**: The strong class imbalance requires more than high accuracy—hybrid prompting improves robustness, with Hybrid GRU_ATT achieving top scores in both metrics.

C. Metric Consideration

Evaluation metrics should match the data distribution and application goals. For example, in imbalanced datasets such as Digital Music and Subscription Boxes, relying solely on accuracy can be misleading, as high scores may simply reflect the dominance of the majority class. In such cases, macro F1-Score provides a more informative and balanced evaluation by equally weighing performance across both positive and negative classes.

To further assess model robustness, we report 95% Confidence Intervals (CIs) for accuracy scores, offering statistical insight into performance variability across runs. These intervals help distinguish meaningful improvements from random fluctuations, especially when comparing models with close accuracy values.

In real-world applications, it is critical to examine not just overall performance but also the types of errors made by the model. False Positives (FP) and False Negatives (FN) may have different consequences depending on the application domain—for example, mistakenly recommending a poor product (FP) versus failing to recommend a good one (FN). To illustrate this, we provide a detailed confusion matrix analysis for the Hybrid GRU_ATT Roberta model, which achieved strong performance across multiple datasets. The breakdown includes precision, recall, true/false positives and negatives, and class-wise F1-Scores. These results, presented in Appendix A (Table A3), offer a concrete example of the model's strengths and remaining challenges, and can help inform future improvements and error mitigation strategies.

D. Best Practice Summary

This summary offers practical advice to align modeling decisions with specific dataset characteristics and performance needs.

- Choose CNN-NonStatic when speed and efficiency are priorities, and performance trade-offs are acceptable.
- Use hand-crafted prompts for quick prototyping or when domain rules are well understood.
- Apply adaptive prompting (GRU_ATT) to tasks involving complex sentence structures or longer texts.
- Opt for hybrid prompting—particularly Hybrid GRU_ATT—for the most consistent and high-performing results across datasets.
- Emphasize the F1 score in evaluations involving class imbalance.

These recommendations are intended to help practitioners align model and prompting choices with the characteristics of their specific data and application goals.

VI. Conclusion

This work presented an in-depth evaluation of ten deep learning models for sentiment classification, including both CNN-based architectures and a range of BERT-based configurations. We examined how various prompting strategies, from no prompting to hand-crafted, adaptive, and hybrid approaches—affect model performance across five datasets.

The results confirm that BERT-based models, especially those incorporating adaptive and hybrid prompting, significantly outperform CNNs in both accuracy and F1-Score. Hybrid strategies that combine structured prompts with learned components, such as Hybrid GRU_ATT, consistently deliver the strongest results, particularly on datasets with informal language or class imbalance.

While these findings are promising, there are limitations. Our analysis was restricted to English-language, binary classification tasks. It remains to be seen how well the proposed approaches generalize to multilingual data or more fine-grained sentiment labels.

Another practical consideration is computational cost. BERT-based hybrid models introduce additional overhead in both memory usage and inference time, which may be a limiting factor in latency-sensitive or resource-constrained environments. Although we recorded training and inference times during our experiments, these measurements were affected by concurrent processes and shared system loads, making them unreliable for consistent comparison. For transparency, the approximate timing results are included in Appendix A (Table A4). A controlled benchmarking setup is needed for an accurate and reproducible evaluation of computational efficiency.

Overall, this study highlights the importance of tailoring both model architecture and prompting strategy to dataset characteristics. In future work, we plan to investigate the adaptability of hybrid prompting across domains and explore more efficient alternatives that balance performance with computational feasibility. The other direction can be to extend this analysis to include other large language models such as XLNet, GPT-4, and LLaMA to evaluate their strengths and weaknesses under different prompting and fine-tuning paradigms. Also, incorporating more diverse datasets, including non-English languages, domain-specific texts, and multimodal data, would further validate and generalize the conclusions drawn from this research.

APPENDIX A: SOFTWARE USED AND ADDITIONAL EXPERIMENTAL RESULTS

TABLE A1: SOFTWARE AND LIBRARY DEPENDENCIES

Library / Tool	Installed Version	Description
Python	3.11.7	Programming language used for all experiments
PyTorch	2.4.1	Core deep learning framework for CNN and BERT models
HuggingFace Transformers	4.42.3	For loading and fine-tuning pretrained models like 'bert-base-uncased'
TorchText	Built-in	Preprocessing and dataset pipeline for CNN models
Scikit-learn	1.2.2	Metrics (accuracy, F1), data splitting, CI calculation
Gensim	4.3.0	For word2vec and fastText embeddings
NLTK	3.8.1	Tokenization, stopword removal, text preprocessing
Matplotlib	3.8.0	For result visualization and accuracy plots
Seaborn	0.12.2	For CI error bars and statistical visualization
Jupyter Notebook	—	Interface used for developing CNN models interactively
Conda	24.1.2	Used for environment and dependency management
Hardware	—	MacBook Pro (14-inch, 2021), M1 Pro chip, 32 GB RAM, macOS Sequoia 15.2

TABLE A2. ACCURACY AND 95% CONFIDENCE INTERVALS FOR ALL MODELS AND DATASETS

Model	Dataset	Accuracy (%)	CI Lower (%)	CI Upper (%)
CNN-Random	SST2	73.63	71.61	75.65
	Tweets	75.34	73.61	77.07
	IMDB	87.8	86.52	89.08
	Sub.Boxes	90.52	89.29	91.75
	Music	93.73	92.6	94.86
CNN-Static	SST2	85.98	84.39	87.57
	Tweets	78.21	76.56	79.86
	IMDB	90.7	89.56	91.84
	Sub.Boxes	92.73	91.64	93.82
	Music	95.3	94.31	96.29
CNN-NonStatic	SST2	86.48	84.91	88.05
	Tweets	78.84	77.2	80.48
	IMDB	90.3	89.14	91.46
	Sub.Boxes	92.94	91.86	94.02
	Music	95.56	94.6	96.52
BERT No Prompt	SST2	78.87	77	80.74
	Tweets	72.95	71.17	74.73
	IMDB	78.43	76.82	80.04
	Sub.Boxes	87.41	86.01	88.81
	Music	92.26	91.01	93.51
BERT Hand-crafted	SST2	80.25	78.42	82.08
	Tweets	73.15	71.38	74.92
	IMDB	81.13	79.6	82.66
	Sub.Boxes	88.43	87.08	89.78
	Music	92.03	90.76	93.3
BERT Adaptive GRU_ATT	SST2	90.77	89.44	92.1
	Tweets	80.2	78.6	81.8
	IMDB	90.55	89.4	91.7
	Sub.Boxes	93.06	91.99	94.13
	Music	95.04	94.02	96.06
BERT Hybrid LSTM	SST2	91.59	90.32	92.86
	Tweets	80.41	78.82	82
	IMDB	91.19	90.08	92.3
	Sub.Boxes	93.06	91.99	94.13
	Music	96.04	95.13	96.95
BERT Hybrid GRU_ATT	SST2	92.2	90.97	93.43
	Tweets	81.5	79.94	83.06
	IMDB	91.35	90.25	92.45
	Sub.Boxes	93.29	92.24	94.34
	Music	96.27	95.38	97.16

TABLE A3. DETAILED CONFUSION MATRIX METRICS AND DERIVED EVALUATION SCORES FOR HYBRID GRU_ATT ROBERTA MODEL

Dataset	TP	TN	FP	FN	Precision	Recall	NPV	TNR	F1_Positive	F1_Negative	Macro_F1
SST2	880	836	76	29	92.05%	96.81%	96.65%	91.67%	94.37%	94.09%	94.23%
Tweets	957	994	219	225	81.38%	80.96%	81.54%	81.95%	81.17%	81.74%	81.46%
IMDB	1187	1160	84	69	93.39%	94.51%	94.39%	93.25%	93.95%	93.81%	93.88%
Sub. Boxes	1493	544	68	61	95.64%	96.07%	89.92%	88.89%	95.85%	89.40%	92.63%
Music	1582	110	31	30	98.08%	98.14%	78.57%	78.01%	98.11%	78.29%	88.20%

TABLE A4. APPROXIMATE TRAINING TIME (IN SECONDS) FOR ALL MODEL VARIANTS ACROSS DATASETS

Dataset	**CNN-Random**	**CNN-static**	**CNN-non-static**	**No Prompt BERT**	**Hand-crafted BERT**	**Adaptive GRU_ATT BERT**	**Hybrid LSTM BERT**	**Hybrid GRU_ATT BERT**	**Hybrid LSTM Roberta**	**Hybrid GRU_ATT Roberta**
SST2	1,526	278	258	18,873	14,756	12,321	11,200	17,571	20,757	22,996
Tweets	2,766	2,470	2,868	26,056	22,788	12,524	10,302	23,434	30,614	35,696
IMDB	3,071	1,751	3,069	38,912	55,059	35,564	42,324	87,696	29,464	39,485
Sub. Boxes	6,720	7,124	7,591	13,364	22,753	33,615	30,191	42,498	29,544	42,506
Music	6,808	3,291	3,620	24,109	30,008	14,771	12,650	22,048	24,965	36,523

Note: These times were recorded during experiments but may be affected by background processes and shared hardware. Reported values are indicative and not normalized across environments.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Bo Huang conducted the research, carried out the experiments, and prepared the initial draft of the manuscript. Dr. Fei Song supervised the research, provided critical guidance throughout the study, and contributed to the review and revision of the manuscript. Both authors had approved the final version.

REFERENCES

[1] L. Zhang and B. Liu, "Sentiment analysis and opinion mining," in *Encyclopedia of Machine Learning and Data Science*, Springer, 2023, pp. 1–13.

[2] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[3] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.

[4] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[5] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," arXiv preprint, arXiv:1510.03820, 2015.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[7] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*, Springer, 2012, pp. 415–463.

[8] B. Pang, L. Lee *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[9] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. the 12th International Conference on World Wide Web*, 2003, pp. 519–528.

[10] S. Behdenna, F. Barigou, and G. Belalem, "Document level sentiment analysis: A survey," *EAI Endorsed Transactions on Context-Aware Systems and Applications*, vol. 4, no. 13, 2018.

[11] B. Yang and C. Cardie, "Context-aware learning for sentence-level sentiment analysis with posterior regularization," in *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 325–335.

[12] A. Ferrari and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering," *Automated Software Engineering*, vol. 26, no. 3, pp. 559–598, 2019.

[13] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proc. the 2nd International Conference on Knowledge Capture*, 2003, pp. 70–77.

[14] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, pp. 347–354.

[15] G. Brauwers and F. Frasincar, "A survey on aspect-based sentiment classification," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–37, 2022.

[16] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.

[17] B. Huang and F. Song, "Empirical study of BERT-Based models for sentiment analysis," in *Proc. the 32nd ACIS International Summer Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD2025-Summer IV)*, Brampton (Greater Toronto Area), Canada: IEEE, July 2025.

[18] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexiconbased methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[19] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," arXiv preprint, arXiv:cs/0205070, 2002.

[20] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[21] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint, arXiv:1408.5882, 2014.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in *Proc. the China National Conference on Chinese Computational Linguistics*, Springer, 2019, pp. 194–206.

[24] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," arXiv preprint, arXiv:2012.15723, 2020.

[25] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," arXiv preprint, arXiv:2104.08691, 2021.

[26] T. Schick and H. Schütze, "Exploiting cloze questions for few shot text classification and natural language inference," arXiv preprint, arXiv:2001.07676, 2020.

[27] S. Varia *et al.,* "Instruction tuning for few-shot aspect-based sentiment analysis," arXiv preprint, arXiv:2210.06629, 2022.

[28] E. Memiş, H. Akarkamçı, M. Yeniad, J. Rahebi, and J. M. Lopez-Guede, "Comparative study for sentiment analysis of financial tweets with deep learning methods," *Applied Sciences*, vol. 14, no. 2, 588, 2024.

[29] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "LLMs and NLP models in cryptocurrency sentiment analysis: A comparative classification study," *Big Data and Cognitive Computing*, vol. 8, no. 6, 63, 2024.

[30] P. Zhang, T. Chai, and Y. Xu, "Adaptive prompt learning-based fewshot sentiment analysis," *Neural Processing Letters*, vol. 55, no. 6, pp. 7259–7272, 2023.