International Journal of Numerical Methods
for Calculation and Design in Engineering

ARTICLE

Enhancing Few-Shot Text Classification with Parameter-Efficient Tuning of Large Language Models

MeiJie Zhao[1], Yi Sun[2,3], Xuhao Fan[4] and Yu Tian[5,*]

[1]Qingdao University of Science and Technology, No. 99 Songling Road, Qingdao, China
[2]School of Microelectronics, South China University of Technology, Guangzhou, China
[3]Guangzhou International Campus, South China University of Technology, 777 Xingye Avenue East, Panyu District, Guangzhou, China
[4]School of Computer Science and Technology, Anhui University, No. 3 Feixi Road, Shushan District, Hefei, China
[5]Department of Computer Science, Wenzhou-Kean University, Wenzhou, China
*Corresponding Author: Yu Tian. Email: yutian4608@outlook.com or tianliudao@gmail.com

Received: 03 December 2025; Accepted: 12 March 2026

ABSTRACT: Traditional few-shot text classification models focus only on label prediction and cannot extract structured information such as entities or events, limiting their usefulness in real-world, semantics-driven tasks. They also rarely use external knowledge or parameter-efficient tuning, leading to shallow representations and weaker performance. To address this, this paper proposes a knowledge-aware multi-task framework that integrates few-shot classification with entity and event extraction. A single BERT encoder with IA^3 adapters enables efficient tuning, while semantic triples extracted via spaCy and aligned with WordNet and ConceptNet are encoded using TransE. A BiLSTM captures sequential context and a softmax decoder performs token-level prediction. Experiments show strong results—97.97% accuracy, 98.00% precision, 97.95% recall, and 97.96% F1—surpassing state-of-the-art baselines. Ablation studies confirm the value of the knowledge-enhanced, multi-step design, demonstrating suitability for low-resource, knowledge-centric applications.

KEYWORDS: Few-shot learning; multi-task learning; text classification; entity and event extraction; knowledge-aware modeling; and bidirectional encoder representations from transformers

1 Introduction

In the past few years, along with the rapid advances of NLP, machines have been enabled to read vast amounts of unstructured text, classify it, and extract structured information from it. Text classification, named entity recognition, and event extraction are key steps while describing an exemplary pipeline for news analysis, social media monitoring, and knowledge base construction [1,2]. On the flipside, these systems require very fine-grained annotation data, heavy fine-tuning of deep models, with such an approach is information and computationally expensive. This presents a major hindrance in a low-resource or real-time environment where very little labeled data exists, and where fast domain adaptation has to be executed [3–5].

The foremost factors limiting the performance of NLP tasks in low-resource settings are the absence of annotated examples, model-specific rigidity for tasks [6–8], and impaired comprehension of contexts. Usually, the models trained in fully supervised settings do not generalize in few-shot settings, especially when the changes in target domains are huge compared to the source, which is the huge domain difference. Finally, treating classification independently of information extraction causes redundancy and inefficiency, leading to fragmented predictions that lack coherence [9,10]. Models also often find it disadvantageous that

they cannot tap several external knowledge sources to build interpretation and reasoning abilities, especially where entity-event associations and world knowledge are implicated.

To handle these challenges, researchers have suggested many standalone solutions in different NLP tasks. For text classification, well-tuned supervised models comprising CNN [11,12], RNN [13], and BERT-like transformer [14,15] encoders usually have good performance on big datasets. For NER (Named Entity Recognition) and event extraction, sequence labeling methods, including Bi-LSTM with softmax taggers and span-based extractors, have gained popularity in the field. At the same time, knowledge graphs such as ConceptNet, Wikidata, and WordNet have been used externally to help downstream reasoning. While promising on their own, these methods often do not synergize with each other, while lacking scalability and adaptability in low-resource or multi-task scenarios.

To overcome these limitations, the integrated and knowledge-aware NLP framework performs both few-shot text classification and joint entity-event extraction using a common, parameter-efficient architecture. Structured prompt templates are generated from the extracted information triples, and label representation is expanded by knowledge graphs to improve semantic reasoning. A shared transformer-based encoder is trained by an efficient adaptation technique called Input-Aware Adapter Adjustment (IA^3) without retraining the entire model. Also, a custom loss function incorporates structural knowledge constraints guiding the model's learning process. This method not only addresses data scarcity but also guarantees coherent information extraction and classification while pushing the boundaries of resource-efficient and knowledge-aware NLP. The primary contributions of the paper are listed below,

- Develop a knowledge-enhanced prompt construction strategy by injecting factual triples extracted using spaCy's dependency parsing and expanding label verbalizers using semantic resources such as WordNet and ConceptNet.
- Implement a parameter-efficient fine-tuning mechanism through Input-Aware Adapter Adjustment (IA^3), significantly reducing the number of trainable parameters while maintaining performance.
- Integrate a hybrid extraction module combining BERT and BiLSTM layers to capture both contextual and sequential dependencies for high-accuracy entity and event detection.
- Formulate a multi-level loss function that incorporates classification loss, Bi-LSTM with softmax-based extraction loss, and TransE-based knowledge alignment, guiding the model towards semantically consistent learning.

The rest of the paper is structured as follows: Section 2 details a comprehensive review of recent developments in few-shot text classification, as well as joint entity-event extraction. Section 3 explicates the knowledge-aware multi-task learning framework, its architecture, prompt construction, and training strategy. Section 4 delineates the experimental setup, performance comparison, and interpretation of results. Section 5 concludes the paper and indicates future work in the hope of making the model even more adaptable and efficient with the use of actual NLP applications.

2 Literature Review

Liu et al. [16] considers the problem of few-shot text classification and proposes two distribution estimation methods, Way-DE and Shot-DE, which use unlabeled query samples to avoid negative transfer problems, thereby considerably peforming improvement for classification across various datasets. The TART network introduced by Lei et al. [17] enhances few-shot text classification by projecting the class prototypes into task-adaptive reference points and applying discriminative reference regularization to maximize the divergence between prototypes, ensuring better generalization and performance on benchmark datasets. Wang et al. [18] suggested a GORAG model which enhances few-shot text classification by constructing an adaptive information graph and using weighted edges to extract reliable information, dynamically retrieving relevant context to ensure classification accuracy when facing data with few labels and changing target label sets.

Hou et al. [19] offers a general FSTC paradigm in which label templates are embedded into input sentences, and supervised contrastive learning methods and attention mechanisms are employed to achieve more discriminative text representations and better performances, especially in the 1-shot setting. Lei et al. recommended an AMGS model [20] that improves few-shot text classification by means of minimizing overfitting via self-supervised tasks and an adaptive meta-learner that distinguishes positive and negative gradients to enhance generalization of the model and performance on unseen tasks. Luo et al. [21] proposed BT-Classifier for prompt-based data augmentation and treats the large-scale language model as a black-box, thus achieving better performance in few-shot classification tasks than state-of-the-art baselines without any fine-tuning of model parameters, ensuring both efficiency and effectiveness.

Kim et al. [22] augment few-shot text classification by generating variously structured syntactic samples horizontally to a certain semantic content. Furthermore, various novel train-val splitting strategies are proposed that optimize performance and hence, demonstrate significant increases in performance across different state-of-the-art classification techniques. Using Multi-level Distributional Signatures Wang et al. [23] captured domain-agnostic, domain-specific, and class-specific features improves few-shot text classification by enhancing domain adaptability and classification across different domains, as proved by the proposed MultiDS in this study. Few-shot text classification was improved upon Li et al. [24] via a novel knowledge-distillation approach that draws self-supervised information from unlabeled samples; and also via a graph aggregation structure to enforce a stronger interaction among the support and query sets and to thereby learn better discriminative representations. Few-shot text classification was improved via the Unified Prompt Tuning (UPT) framework by Wang et al. [25], which captures unified prompting semantics from heterogeneous source tasks, uses a Prompt-Options-Verbalizer paradigm, and enhances generalization via a Knowledge-enhanced Selective Masked Language Modeling task.

Problem Statement

Traditional text classification methods often fail in few-shot learning scenarios due to the severely limited labeled training data and inability to grasp more profound semantic relationships [26]. Furthermore, all prevailing methods treat classification and information extraction as separate tasks, thereby failing to provide structured knowledge like entities and events, which are paramount from the standpoint of real-world applications. Most approaches also do not consider the integration of external knowledge sources and thereby fall short of providing adequate contextual understanding. Apart from this, the parameter inefficiency of large language models obstructs adaptability in low-resource scenarios. Consequently, there must be one framework, parameter-efficient and knowledge-aware, that performs few-shot classification in conjunction with entity-event extraction, thereby drawing on semantic information from the outside and improving under data-scarce settings.

3 Proposed Methodology

The framework presented in the current study deals with few-shot text classification and joint entity-event extraction to arrive at a knowledge-aware, parameter-efficient solution constructed via multi-task learning. The input text gets fully preprocessed, going through normalization, tokenization, and semantic

enhancement through triple extraction using Spacy techniques. These extracted triples are then embedded as MLM prompts to create a better contextual view. The prompt construction module further elaborates verbalizer expansion through external knowledge graphs like WordNet or ConceptNet to permit a better label generalization in low-resource scenarios. To adapt efficiently to the domain while updating the lowest possible parameters, a shared BERT transformer encoder equipped with IA^3 sits at the core. The encoder feeds two parallel task heads, one for masked token classification and the other for entity-event extraction with a BERT–BiLSTM pipeline. The system is trained by using a weighted multi-objective loss, synthetically combining cross-entropy for classification, Bi-LSTM for extraction, and TransE-based knowledge alignment for semantic coherence. Final output includes class labels, BIO-tagged entities and events, semantic triples, and the confidence scores, formatted both for display and machine consumption. This comprises a unified pipeline that can address the data scarcity and semantic complexity issues latent in the low-resource NLP settings. The proposed workflow is shown in Fig. 1.

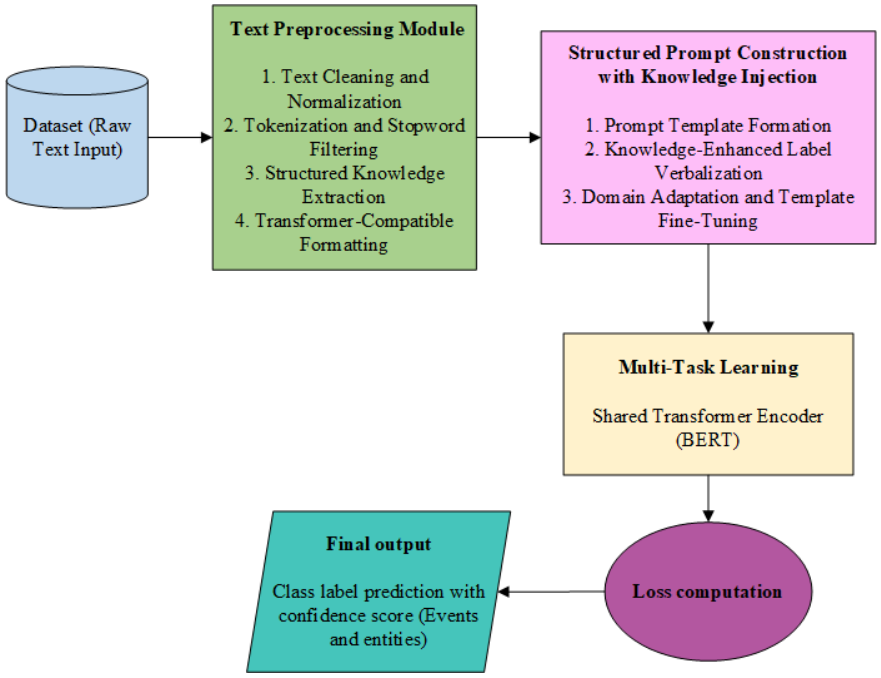

Figure 1: Knowledge-Aware Multi-Task Architecture for Few-Shot Classification and Entity-Event Extraction.

3.1 Preprocessing Stage

The preprocessing stage forms the basis of the proposed framework and transforms unstructured raw text into a cleaned and semantically enriched format suitable for knowledge-guided prompt learning and multi-task transformer-based modeling. It traverses from traditional text cleaning to structured knowledge extraction and transformer-compatible formatting. The result is a text from a linguistic and logical point of view that unfolds into classification and extraction tasks in unison.

3.1.1 Text Cleaning and Normalization

This first step involves text normalization of the raw input. It converts all characters to lowercase, strips punctuation and special characters, standardizes common symbols (e.g., changing "&" into "and") and removes any excessive space characters. This kind of normalization is paramount for reducing the diversity of the lexica and increasing the consistency of the whole dataset. Formally, given a text x, its normalized form is x'.

$$x'\mathcal{T}_{\text{norm}}(x) = \text{lowercase}(\text{replace\_symbols}(\text{strip}(x))) \quad (1)$$

where x represents the original input sentence and $\mathcal{T}_{\text{norm}}$ denotes the composite normalization function.

3.1.2 Tokenization and Stopword Filtering

After normalization, two levels of tokenization are applied to the input text. The first level of tokenization is the classical word-level tokenization, performed through spaCy, allowing for stopword removal and optional lemmatization. At a second level of tokenization, a subword tokenizer of the Byte-Pair Encoding type, as implemented in BERT, is used to guarantee that tokenization is consistent with transformer-based modelling. Classical tokenization digests some common stopwords, those that add very little or no semantic discernment at all (e.g., "is", "the", "on"), among others. The stop word removal operation is defined as:

$$\mathcal{T}_{\text{sw}}(x') = \{w_i \in x' \mid w_i \notin \mathcal{S}\} \tag{2}$$

where $\mathcal{S}$ denotes the set of predefined stopwords and w_i represents individual tokens.

3.1.3 Structured Knowledge Extraction

To combine the model with factual semantics, the cleaned text is processed by spaCy's dependency parser. The first phase consists of extracting (head, relation, tail) triples of the form (h, r, t), which represent subject-predicate-object constructs within the sentence. These triples are then plugged into structured prompt templates and eventually fed into the knowledge-guided loss. In addition, dependency parsing with spaCy can be carried out to better assign roles between detected entities and event triggers to improve syntactic grounding on downstream tasks.

3.1.4 Transformer-Compatible Formatting

The last step arranges the processed data to be modeled by the transformer architecture. Special tokens such as [CLS], [MASK], and [SEP] are inserted to delimit the structure needed by models like BERT. Attention masks and segment IDs are also generated. In mathematical terms, the entire preprocessing pipeline can be viewed as a composition of transformation functions:

$$\mathcal{T}(x) = \mathcal{T}_{\text{format}}\left(\mathcal{T}_{\text{token}}\left(\mathcal{T}_{\text{sw}}\left(\mathcal{T}_{\text{norm}}\left(x\right)\right)\right)\right) \tag{3}$$

Where $\mathcal{T}$ represents the full preprocessing function applied to input x.

3.2 Structured Prompt Construction with Knowledge Injection

This semi-structured prompt module is designed to convert cleaned sentences into template-guided, semantically informative prompts that fit the Masked Language Modeling (MLM) paradigm. Based on the pseudo-log-likelihood scoring framework, this step uses token-level masking and contextual knowledge to steer classification and information extraction in few-shot learning settings shown in Fig. 2.

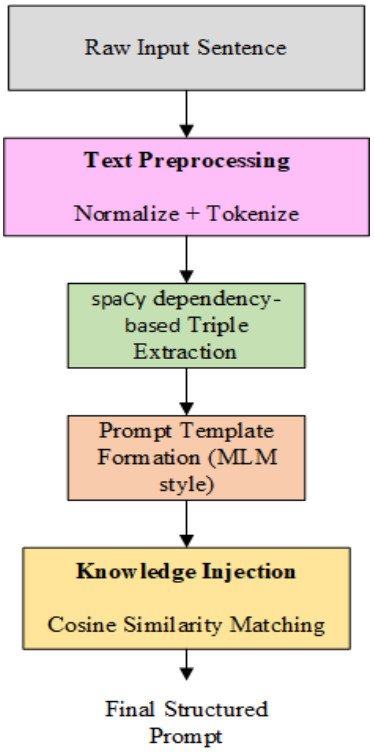

Figure 2: Prompt Construction and Knowledge Injection Flow.

3.2.1 Prompt Template Formation via *Masked Modeling*

Each sentence is embedded into the prompt structure suitable for MLMs such as BERT in the process of preprocessing. The idea is to mask a token, usually a label or an event trigger, and predict it given the bidirectional context. Let a prompt P be constructed as:

$$P = [\text{CLS}]\ \text{Entity}_{\ 1}\ \text{Relation Entity}_{\ 2}.\ \text{It is}\ [\text{MASK}]\ [\text{SEP}] \tag{4}$$

The model expects the classification label at '[MASK]' (e.g., topic, intent) to be inferred in relation with the remaining context surrounding the sentence, which may or may not contain factual information. For this purpose, the model computes the pseudo-log-likelihood (PLL) scores by sequentially masking each token in the sentence and adding the log probabilities for that token:

$$\text{PLL}(P) = \sum_{t=1}^{|P|} \log P_{\text{MLM}}(w_t \mid P_t) \tag{5}$$

where P_t denotes the prompt with the t^{th} token masked. This formulation supports unsupervised scoring and label prediction, leveraging deep bidirectional context.

3.2.2 Injection of spaCy Dependency

The factual triples (h, r, t) extracted using spaCy's dependency-based triple extraction, which are directly embedded in the prompt template with the objective of enhancing semantic richness. These triples provide scaffolding to the knowledge base and help reduce ambiguity, especially under few-shot conditions. For example:

- Triple: ("WHO", "declared", "pandemic")
- Prompt: [CLS] WHO declared pandemic. It was [MASK] [SEP].

This technique allows the MLM to resolve [MASK] to appropriate labels like "health" or "emergency" based on factual priors and contextual fluency.

3.2.3 Knowledge-Enhanced Label Verbalization (Cosine Similarity Matching)

The use of a single class token (e.g., "politics") at the [mask] position is often found to damage the robustness of the model in few-shot setups. To rectify this, this study convert a class label C into a verbalizer set $\mathcal{V}_C$ consisting of semantically-related words using a lexical knowledge base like WordNet, ConceptNet, or RelatedWords.io. Candidate tokens are chosen according to cosine similarity in the embedding space. Thus, candidate token v_i gets added to the verbalizer set if:

$$\cos\left(\vec{v}_i, \vec{C}\right) \geq \theta \tag{6}$$

where $\vec{v}_i$ is the embedding of the candidate token, $\vec{C}$ is the embedding of the class label, $\theta \in [0, 1]$ is a similarity threshold. The resulting verbalizer set is:

$$\mathcal{V}_C = \left\{v_i \in \mathcal{V}_{\text{all}} \mid \cos\left(\vec{v}_i, \vec{C}\right) \geq \theta\right\} \tag{7}$$

This approach improves label coverage and supports flexible decoding: even if the model predicts a synonym (e.g., "government"), it can still be mapped back to the class "Politics" based on similarity to the verbalizer set $\mathcal{V}_C$.

3.2.4 Domain Adaptation and Template Fine-Tuning

Adaptation to domain can improve the MLM for final tasks. In-domain text with dynamic masking is optionally used for adaptation of BERT. While prompt templates are fine-tuned on development data by maximizing the pseudo log-likelihood over a sample set of domain-specific sentences:

$$\mathcal{J}_{\text{PLL}}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \text{PLL}(P_i; \Theta) \tag{8}$$

where Θ is a set of model parameters, and P_i are masked prompts in the training corpus. After forming the knowledge-enriched prompt, the prompt is tokenized via a BERT-compatible tokenizer to get input IDs, attention masks, and position embeddings. The encoded sequence vectors are then sent to the shared encoder for dual-task prediction (classification and extraction). The MLM objective will make certain that every masked token is predicted using complete left and right context; in so doing, it brings about syntactic and semantic fluency.

3.3 Multi-Task Learning Architecture

At the center of the framework sat our unified multi-task-learning architecture that supports few-shot text classification and joint entity-event extraction simultaneously. By sharing a transformer encoder among tasks and having discrete output heads per task, the architecture facilitates parameter efficiency, semantic coherence, and strong generalization in low-supervision environments.

3.3.1 Shared Transformer Encoder

Structured prompts are prepared and then tokenized using a pre-trained model like BERT, which is acting as an encoder. This encoder is going to capture deep contextual representations from both directions, thus proving to be quite efficient for both classification and sequence labeling tasks. Let $X = \{x_1, x_2, \dots, x_n\}$ be the input, then the encoder produces contextual embeddings $H \in \mathbb{R}^{n \times d}$ as,

$$H = \text{BERT}(X) \tag{9}$$

where d is the hidden size of the model. These embeddings are then routed to two separate task-specific heads for classification and extraction.

3.3.2 Classification Head for Few-Shot Learning

The model uses the masked token task head to perform few-shot classification. The [MASK] token position m is found in the input, and its corresponding hidden representation h_m is passed through a softmax layer on the verbalizer vocabulary $\mathcal{V}$ to predict the class label.

$$P_{\text{class}}(y \mid h_m) = \text{softmax}(W_c h_m + b_c) \tag{10}$$

where $W_c \in \mathbb{R}^{|\mathcal{V}| \times d}$ and b_c are the trainable classification parameters. The predicted token is then mapped back to its corresponding class using a predefined verbalizer.

3.3.3 Extraction Head for Entity and Event Detection

The extraction head in the proposed framework is designed to detect and label semantic components within input sequences; especially the so-called named entities (persons, organizations, or locations), and event triggers (actions or incidents). These components are important to be studied later for downstream tasks such as working with knowledge graphs, relation extractions, and question answering. To provide stronger extraction mechanisms and more accurate extractions, a hybrid architecture combining BERT and Bi-LSTM networks was adopted, followed by a softmax classification layer. Thus, the transformer-based contextual encoding capability is complemented by the sequential modeling ability of recurrent networks.

The BERT model acts as an encoder and processes the entire input sequence using its deep bidirectional attention mechanisms. Each token of the input sentence is contextualized with respect to all other tokens, that is, syntactic and semantic dependencies. Formally, for an input sequence $= (x_1, x_2, \dots, x_n)$, BERT provides a sequence of hidden states

$$\text{H} = \text{BERT(x)} = (\text{h}_1, \text{h}_2, \dots, \text{h}_n) \tag{11}$$

These embeddings $\text{h}_i \in \mathbb{R}^d$ provide rich contextual information per token. Though BERT has powerful global dependencies, it does not explicitly model the sequential transitions or temporal coherence across tokens, which are crucial for structured tasks like sequence labeling. To incorporate this aspect, the BERT outputs are fed into a Bi-LSTM layer, which processes local and long-range dependencies in both directions: forward and backward. This is especially good for representing label transitions and the advancement patterns of entities across multi-token spans. At each time step t, the Bi-LSTM takes the input as follows:

$$\vec{h}_t, \overleftarrow{h}_t = \text{BiLSTM}(\text{h}_t) \text{ and } \text{z}_t = \left[\vec{\text{h}}_t; \overleftarrow{h}_t\right] \tag{12}$$

The output sequence $\mathbb{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ thus embedded the contextual semantics and temporal ordering of tokens. Each output vector $\mathbf{z}_t \in \mathbb{R}^{2h}$ is fed into a softmax classification layer that predicts a distribution over a predefined tag space (e.g., BIO tagged labels). The linear projection yields token scores:

$$\mathrm{S}_t = \mathrm{W} \cdot \mathrm{z}_t + \mathrm{b} \text{ and } P(y_t \mid \mathrm{z}_t) = \mathrm{softmax}(\mathrm{s}_t) \tag{13}$$

where $\mathrm{W} \in \mathbb{R}^{k \times 2h}, \mathrm{b} \in \mathbb{R}^k$, and k is the number of label classes. The class predicted for each token y_t is simply the label with the maximum probability under this model, and the process is capable of labeling entities and triggers with typical BIO-style tags (e.g., B-PER, I-LOC, O). From here, one can solve overlapping and nested entities due to the Bi-LSTM being fully aware of the sequence. The objective for this part of the pipeline is to minimize the token-wise cross-entropy loss over the whole sequence.

$$\mathcal{L}_{\text{extract}} = -\frac{1}{n}\sum_{t=1}^{n} \log P(y_t^* \mid z_t) \tag{14}$$

where y_t^* denotes the ground-truth tag of the *t*-th token. Such a formulation ensures that each token's contribution to the total loss is proportional to that token's confidence in classification, promoting robust label assignments in the presence of noise or ambiguity.

Moreover, its flexibility gives way to easy multilingual or domain-specific adaptations just by plugging in or fine-tuning another BERT backbone. This same architecture is employed for both named entity recognition and event trigger detection, either by using one combined tag set or by branching one tag set into separate output layers. This reduces parameter overhead, while encouraging inter-task.

3.3.4 Task Integration and Multi-Objective Loss

The multi-task architecture supports simultaneous optimization of two interrelated objectives like few-shot text classification through masked language modeling and joint entity and event extraction through a BIO-based sequence labeling. A multi-objective loss function is used for training such that both tasks are effectively and coherently trained, sharing semantic representations while retaining specific supervision signals from the task.

i) Classification Loss: Masked Token Prediction

For classification, the model is trained to predict the correct class token at the [MASK] position given the input prompt, using a softmax classifier. Let $h_m \in \mathbb{R}^d$ be the hidden representation at the masked token position, and $\mathcal{V}$ be the set of verbalizer tokens denoting all class labels. The predictive probability distribution over the verbalizer vocabulary is computed as follows:

$$P_{\text{class}}(y \mid h_m) = \mathrm{softmax}(W_c h_m + b_c) \tag{15}$$

The classification loss is the standard cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{|\mathcal{V}|} y_i \log \hat{y}_i \tag{16}$$

where $y_i \in \{0, 1\}$ is the one-hot encoded ground truth label, and $\hat{y}_i$ is the predicted probability of class i.

ii) Extraction Loss: A Token-Level Softmax Classifier-Based Sequence Labeling

Extraction refers to the process of assigning BIO-style labels to token sequences in the input sentence so that relevant named entities and event triggers can be identified. This model utilizes the token-based softmax classifier in prediction with k labels. An unnormalized score vector is produced for each token with the output of the Bi-LSTM layer being $\mathbf{z}_t \in \mathbb{R}^{2h}$ at token position t.

$$\mathrm{S}_t = \mathrm{W} \cdot \mathrm{z}_t + \mathrm{b} \tag{17}$$

where $W \in \mathbb{R}^{k \times 2h}$ is the learnable weight matrix and $b \in \mathbb{R}^{k}$ is the bias vector. The predicted probability distribution over the tag space is computed using the softmax function:

$$P(y_t = c \mid z_t) = \frac{\exp\left(s_{t,c}\right)}{\sum_{j=1}^{k} \exp\left(s_{t,j}\right)} \tag{18}$$

where $s_{t,c}$ is the score for class c at time step t, and k is the total number of label classes. Predictions are optimized by means of token-level cross-entropy loss. This loss is computed for each prediction at ground truth label y_t^* for the given position.

$$\mathcal{L}_{\text{extract}} = -\sum_{t=1}^{n} \log P(y_t^* \mid z_t) \tag{19}$$

This formulation eliminates the necessity of transition scoring and leads to an easier and faster decoding algorithm based on the token maximum probability. The model predicts directly the most probable label at each position:

$$\hat{y}_t = \arg\max_{c} P(y_t = c \mid z_t) \tag{20}$$

In the concurrent settings of multi-task and few-shot learning, this method stands out practically by computational efficiency and being a perfect match with transformer architecture.

iii) Total Loss Function: Weighted Multi-Task Objective

The final training objective is a weighted combination of the classification and extraction losses:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{extract}} \tag{21}$$

Here, α *and* β are scalar hyperparameters balancing the contribution from each task. The term $\mathcal{L}_{\text{cls}}$ is applied to encourage the accurate prediction of labels in few-shot classification with an MLM-style prompt. $\mathcal{L}_{\text{extract}}$ implements structured sequence tagging of entities and events through Bi-LSTM-based decoding.

This joint purpose promotes the learning of unified, generalizable representations that are semantically rich and task-aware. Moreover, the design provides flexible tuning depending on available data, such as emphasizing few-shot classification when extraction data is scarce or *vice versa*. By optimizing classification and extraction jointly via the composite objective function, the model learns to correlate intent at the sentence level with token semantics. This training unifies both tasks, which encourages less overfitting in low-resource setups and supports a coherent interpretation of both label and span outputs in downstream NLP applications.

3.4 Parameter-Efficient Fine-Tuning with IA³

Large-scale language models like BERT perform very well across all NLP tasks; however, such models are too expensive to fine-tune in their entirety and are very memory-intensive as well. This framework introduces a parameter-efficient fine-tuning approach called Input-Aware Adapter Adjustment (IA^3). With this technique, models can adapt to tasks and domains by adjusting a very small number of extra parameters. This greatly reduces the cost of training without sacrificing performance in multitask and few-shot settings.

3.4.1 Motivation for Parameter-Efficient Learning

Traditional fine-tuning updates each parameter of a pre-trained model Θ, which may become infeasible as models reach hundreds of millions of parameters, especially in resource-constrained environments. Full fine-tuning, on the downside, tends to catastrophic forgetting and can reduce the ability to generalize over multiple tasks. To lessen such drawbacks, IA^3 injects input-aware scaling vectors in each

transformer layer, a minimal yet effective mechanism to steer adaptation without interfering with the model's pre-learned weights.

3.4.2 IA³ Integration into Transformer Layers

The IA^3 method is incorporated into each transformer block by attaching learnable scaling vectors with the key (K), value (V), and feed-forward Network(FFN) components. This approach ensures that the activations are modified according to input-specific properties, without altering the original backbone weights. Formally, the attention and feed-forward outputs in a given transformer layer after modification are expressed as follows:

$$K' = K \odot \gamma_k, V' = V \odot \gamma_v, \mathrm{FFN}' = \mathrm{FFN}(H) \odot \gamma_f \tag{22}$$

where $\odot$ denotes element-wise multiplication, H is the input to this layer, and $\gamma_k, \gamma_v, \gamma_f \in \mathbb{R}^d$ are the trainable IA^3 vectors corresponding to the key, value, and feed-forward modules, respectively. These are very lightweight trainable vectors that require only 3d additional parameters per layer and are sufficient to steer the model's behavior under new conditions.

3.4.3 Application in Multi-Task Few-Shot Settings

In the proposed framework, IA^3 is coherently applied to all transformer layers in the shared BERT encoder. This gives the model flexibility to adapt both classification and extraction tasks at low computational cost. During training, only the IA^3 scaling vectors and task-specific heads (classification or extraction) are updated while the backbone model remains frozen. This drastically reduces the number of trainable parameters and speeds up convergence, a huge merit in few-shot learning scenarios where training data is scarce. Let $\Phi \subset \Theta$ denote the IA^3 parameter set and task heads. The updated objective function becomes

$$\min_{\Phi} \mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{extract}} \tag{23}$$

subjected to the constraint that all other parameters in $\Theta \setminus \Phi$ are frozen. The combination of IA^3 upgrades the proposed framework to fine-tune a large shared encoder for many tasks at once, with an infinitesimally small amount of its parameters. This input-aware tuning thus allows the efficient, scalable, and robust adaptation to new domains and tasks, making it highly successful in few-shot and joint NLP settings.

3.5 Knowledge-Aware Loss Computation

The proposed framework integrates linguistic context, entity structure, and external factual knowledge to maximize the accuracy of prediction and semantic consistency. The ideal method to supervise the model for classification as well as extraction tasks is concurrent at multiple objective levels: these comprise conventional task objectives coupled with supplementary constraining forces on class inference from knowledge-based constraints derived from factual triples. The consideration of these will lead the model to develop knowledge-aware representations convenient for the generalization of this knowledge.

3.5.1 Classification Loss: Masked Token Prediction

The classification task is modeled as an MLM problem; here the model has to predict the correct class token at the [MASK] position under a structured prompt. If $h_m \in \mathbb{R}^d$ denotes the contextualized representation at the masked position and $\mathcal{V}$ is the verbalizer vocabulary, then the distribution over the class labels is defined as follows:

$$P_{\text{class}}(y \mid h_m) = \mathrm{softmax}(W_c h_m + b_c) \tag{24}$$

The classification loss is defined using the standard cross-entropy objective:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{|\mathcal{V}|} y_i \log \hat{y}_i \tag{25}$$

where y_i is the ground truth label indicator and $\hat{y}_i$ is the predicted probability of the label i.

3.5.2 Extraction Loss: Bi-LSTM Output

The extraction task consists of sequence labeling of tokens by BIO tags, with the purpose of identifying named entities and event triggers. The token embeddings generated by a BERT encoder are first passed through a Bi-LSTM layer and then linearly projected onto tag logits.

$$\mathcal{L}_{\text{extract}} = \text{BiLSTM}(H), S = W_e H' + b_e \tag{26}$$

This ensures globally optimal tag decoding via the Viterbi algorithm and enforces sequence-level structural constraints.

3.5.3 TransE-Based Knowledge Embedding Loss

To adapt the predictions further to real-world semantics, TransE-style relational embedding-based knowledge loss is incorporated. Each factual triple (h, r, t) extracted from spaCy is embedded in such a manner that the relational vector translates the head to the tail:

$$h + r \approx t \tag{27}$$

The triple loss encourages valid triples to be closer in embedding space than corrupted (negative) ones:

$$\mathcal{L}_{\text{triple}} = \sum_{(h,r,t)\in\mathcal{T}} [\gamma + \|h + r - t\|_2^2 - \|h' + r - t'\|_2^2]_+ \tag{28}$$

where γ is the margin, and (h', t') is a negative sample.

The squared L2 norm is employed to measure the distance between the translated head embedding and the corresponding tail embedding in the TransE formulation. Compared to the L1 norm, the squared L2 distance ensures smoother gradient behavior and improved differentiability, which contributes to more stable optimization during training. Furthermore, it penalizes larger embedding deviations more strongly, promoting tighter alignment of valid triples in the embedding space and supporting stable convergence within the joint multi-task learning framework. In addition, the continuous and quadratic nature of the squared L2 function provides consistent gradient magnitudes across embedding updates, reducing abrupt parameter fluctuations during backpropagation. This characteristic is particularly beneficial in multi-objective optimization settings, where classification, extraction, and knowledge alignment losses are optimized simultaneously.

3.5.4 Total Loss Function

The total loss function is a weighted sum of the classification, extraction, and knowledge alignment objectives:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{extract}} + \lambda \cdot \mathcal{L}_{\text{triple}} \tag{29}$$

Here, the parameters $\alpha, \beta, and\ \lambda$ belong to $\mathbb{R}_+$ and control the relative weight assigned to each loss component. This knowledge-aware loss design encourages coherent training across levels: classification at the sentence level, extraction at the token level, and structural alignment at the knowledge level. By combining task-specific supervision with relational consistency constraints, the model improves its generalization capacity, especially in few-shot and semantically complex circumstances.

3.6 Output Generation and Prediction Decoding

The final step under the framework is the conversion from the internal outputs of the model into structured human itself capable of interpretation, including inference of class labels from masked token predictions, and structured entity/event sequences decoded from BIO-tag sequences. This stage ensures the application-ready nature of model outputs to further downstream use in knowledge graphs, reports, or intelligent agents.

3.6.1 Class Label Decoding with Cosine Similarity Matching

The few-shot classification head predicts an output token at the [MASK] position in a prompt-formatted input. First, an input is passed through the language model up to the [MASK] token, yielding hidden states $h_m \in \mathbb{R}^d$. After that, h_m is passed to a softmax classifier that calculates a distribution over the verbalizer tokens.

$$P_{\text{class}}(y \mid h_m) = \text{softmax}(W_c h_m + b_c) \tag{30}$$

For improved few-shot generalization, instead of using one token to represent a class label C, a set of semantically related tokens (verbalizers), from external resources such as WordNet, ConceptNet, or RelatedWords.io, is used. Let $\mathcal{V}_C = \{v_1, v_2, \ldots, v_k\}$ be the verbalizer set for class C.

In order to identify the class of a predicted token v^*, the framework computes the cosine similarity between the predicted token embedding $\vec{v}^*$ and the centroid of verbalizer embeddings for each class.

$$\text{Sim}(v^*, C) = \cos(\vec{v}^*, \mu_C), \text{ where } \mu_C = \frac{1}{k}\sum_{i=1}^{k} \vec{v}_i, v_i \in \mathcal{V}_C \tag{31}$$

The predicted class $\hat{C}$ is chosen as the class whose verbalizer set centroid has the highest cosine similarity to the predicted token:

$$\hat{C} = \arg\max_{C \in C} \text{sim}(v^*, C) \tag{32}$$

In the event that a synonym of the exact class label is predicted at [mask], robust label mapping is performed. This lowers classification reliability in low-resource environments due to its reliance on matching in a knowledge-rich semantic space.

3.6.2 Sequence Tag Decoding for Entity and Trigger Extraction

Sequence labeling outputs a BIO-tagged sequence $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$, where each $\hat{y}_t$ denotes the tag attached to the token. The predictions are then decoded through the Viterbi algorithm on the Bi-LSTM layer, enforcing global label consistency.

$$\hat{Y} = \arg\max_{Y \in \mathcal{Y}} P_{\text{tag}}(Y \mid H') \tag{33}$$

Named entities or event triggers are determined by the contiguous spans with matching B- and I-tags. Overlapping multitoken entities are resolved by highest-scoring tag transitions from B-LSTM decoding.

3.6.3 Trigger-Role Mapping and Argument Structuring

Detected triggers undergo the extraction of candidate entities through a syntactic analysis involving dependency grammar or simply via proximity-based heuristics. This is shown in the following example:

"Trump supporters stormed the Capitol."

- Trigger: stormed → ATTACK
- Agent: Trump supporters → PERSON
- Location: Capitol → LOCATION

This mapping can be structured into labeled argument roles, forming complete event frames.

3.6.4 Triple Reconstruction

After entities and event triggers have been extracted, the model presents this information in semantic triples of the form (Entity_1, Trigger/Relation, Entity_2). Such triples express something about the core meaning of the input sentence and can be especially useful in subsequent tasks pertaining to question answering, semantic search, or knowledge graph construction. Thus, the reconstruction is basically pairing the identified entities with the event trigger or relation that is relevant to semantically linking such entities within the sentence. For example, from the sentence "Trump supporters stormed the Capitol", the model extracts:

- Entity 1: "Trump supporters"
- Trigger: "stormed"
- Entity 2: "Capitol"

This represents the structure for the triple (Trump supporters, stormed, Capitol), exactly capturing the subject-action-object semantic nature of symbolic representation and serving what knowledge bases employ in structuring factual data, which can, in turn, be used by intelligent systems to infer new facts or answer electing queries from a given context.

3.6.5 Confidence Scoring and Thresholding

To ensure interpretability and reliability, confidence values are assigned to each predicted output. For classification tasks, confidence is the softmax probability of the token chosen at the [MASK] position, reflecting a degree of preference for the selected label with respect to others. Confidence score estimation is carried out for instances of both the entity and event extraction tasks from the softmax probability given to each label at a token level, with the confidence score prediction for each token being the maximum softmax class probability output by the softmax layer following the BiLSTM decoder.

With semantic triples, combined confidence scores are defined by the aggregation of confidence levels from their trigger word and their entity arguments. The combined confidence score is hence a single scalar value expressing the level of certainty from the model about the extracted relational structure. Confidence scores can be further thresholded to suppress uncertain predictions. This is a typical case in real-world instances such as automated information extraction pipelines, where errors in extraction translate into erroneous conclusions. By restricting the outputs to those with high confidence, the system achieves both precision and reliability.

3.6.6 Output Formatting

Integration with outside systems and human analysis is facilitated in the final decoded outputs by means of structured, machine-readable, and human-readable formats. From a front-end application point of view, beyond mere API consumption, these outputs are given in JSON for easier integration with web services, dashboards, and real-time monitoring tools.

If the output is meant for internal NLP pipelines or annotation tools, it can be rendered as a text with inline annotations, marking entities and events with colours or brackets. This is particularly useful during development or when visualizing. Knowledge-centred applications, such as semantic web interfaces or graph databases, export the structured information as RDF (Resource Description Framework) triples or in a tabular form digestible by means of SPARQL. This allows the knowledge extracted to be ingested into massive knowledge graphs engaged in reasoning, inference, and entity resolution.

4 Results and Discussion

The proposed model was developed on the Python platform. It achieves consistent and reliable performance across classifying and extractive tasks. The balanced metric trends and distinct class-wise precision-recall curves confirmed the varied sequential selection process between relevant and irrelevant frames, proving the efficacy of the model. From the confusion matrix, we see very few misclassifications,

implying almost zero errors for most of the class labels. Training and validation curves show smooth convergence and attest to the fact that no overfitting occurred.

4.1 Dataset Description

AG-News text classification is a strongly recognized benchmark dataset for the text classification tasks, coming from a huge corpus of more than a million news articles collected by the academic search engine ComeToMyHead. Curated by Xiang Zhang, the dataset focuses on four key news categories: World, Sports, Business, and Science/Technology. They have 30,000 training examples and 1900 testing examples per class, thus tallying up to 120,000 training and 7600 testing instances. Each sample consists of a news title and a brief description to allow models to learn context and topical cues. The dataset is presented in the form of CSV files with class labels (1–4), titles, and descriptions. For easy use, it has been considered for natural language processing applications like topic classification, information retrieval, and supervised learning experiments. Its clean and balanced nature makes it useful for low-resource and few-shot performance evaluation scenarios [27].

4.2 Overall Performance of the Proposed Model

Table 1 gives a summary of the key performance metrics of the proposed model, which indicates the model's general applicability.

Table 1: Overall performance of the proposed model.

Metric	Accuracy	Precision	Recall	F1 Score	FNR	FPR
Value	0.9797	0.99	0.9795	0.9796	0.0203	0.0068

The model has achieved an accuracy of 0.9797, meaning that nearly 98% of the predictions pertaining to all the classes generated by the model are correct. Precision of 0.99 indicates that most of the positives predicted by the model are held to be true, while a recall of 0.9795 indicates that the model is able to grasp most existing instances. The F1 score as harmonic mean of precision and recall, 0.9796, also confirms that the model equally balances both. Lastly, since the false negative rate (FNR) of 0.0203 and false positive rate (FPR) of 0.0068 are very small, the model very rarely misses classifying an actual positive or misclassifies an actual negative.

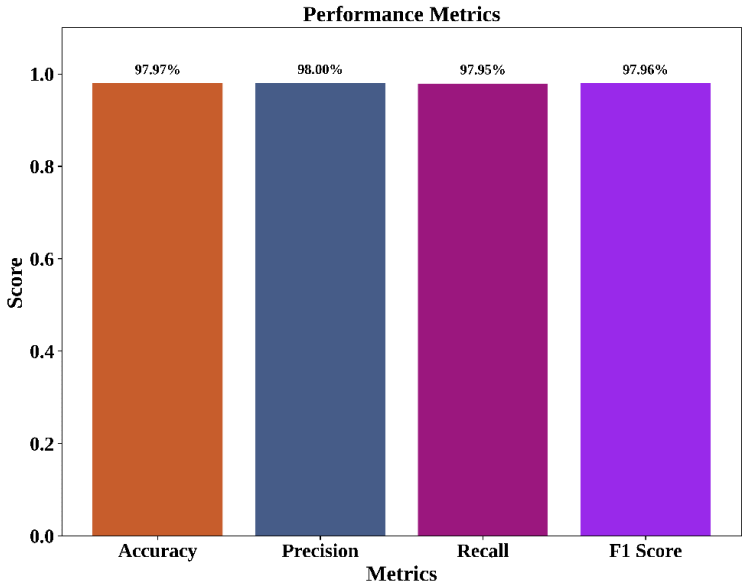

Figure 3: Performance Metrics.

Fig. 3 visually presents the results of a model's classification evaluation using four main metrics: Accuracy, Precision, Recall, and F1 Score. The model performs well, having Precision as the leader at 98.00%, meaning almost all instances predicted as positive were positive. Accuracy is a close second: 97.97% of all classifications-positives and negatives alike-were considered correct. Recall, indicating how well the model identifies all of the pertinent instances, stands at 97.95%, and the closely related F1 Score, the harmonic mean between Precision and Recall, is 97.96%. The closeness of these values tells that the model possesses a good balance between precision and recall, which lends to the overall strength and dependability of the classification task being handled.

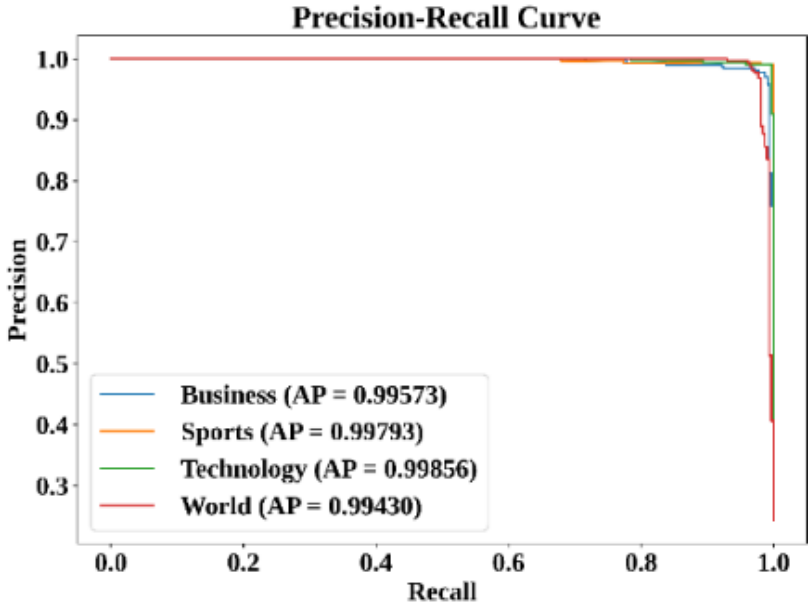

Figure 4: Precision-Recall (PR) Curve.

Fig. 4 measures the performance of a multi-class classifier across four class labels: Business, Sports, Technology, and World. Each curve portrays the tradeoff between precision and recall for a specific class, whereas the Average Precision (AP) scores the area under each PR curve. Extremely high AP scores are achieved by the model for all classes: Business (0.99573), Sports (0.99793), Technology (0.99856), and World (0.99430). Such scores signify an almost perfect ability to differentiate between relevant and irrelevant instances for each class. Given that the curves are almost superimposed near the top right corner of the plot, low precision is hardly ever observed even at relatively high recall levels. This indicates that the model can consistently and strongly classify objects belonging to similar classes into distinct topics. Thus, the model is good at retrieving almost all relevant entities while keeping the number of false positives very low.

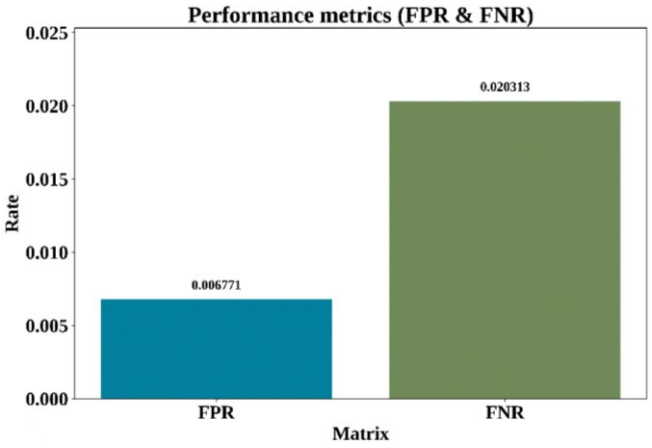

Figure 5: Performance metrics (FPR & FNR).

Fig. 5 portrays the error rates of the model with respect to False Positive Rate and False Negative Rate. The 0.006771 FPR is a very small number, implying that false alarms, i.e., negative cases incorrectly classified as positive, are rarely generated by the model. The slightly higher FNR of 0.020313 means perhaps a few positive cases are wrongly ignored. The two error rates are both fairly low, signaling that the model is reliable and possesses strong discriminatory abilities. The chart reinforces the claim that the classifier is extremely precise and carries an acceptable recall, maintaining very minimal misclassification rates on either end.

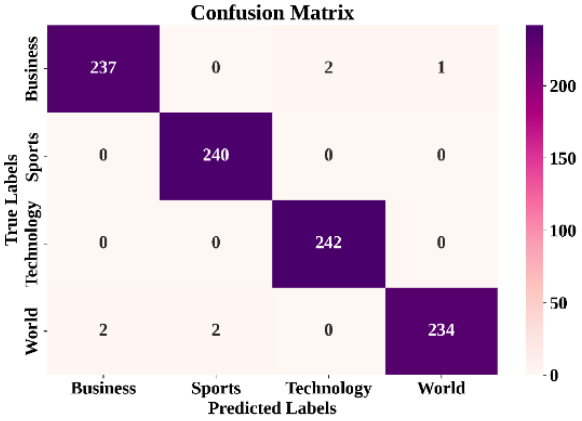

Figure 6: Confusion matrix.

The confusion matrix presented in Fig. 6 offers fine detail into view regarding the classification performance of the model over these four categories: Business, Sports, Technology, and World. Each cell reflects the number of predictions made for any given combination of true and predicted labels. All four features in the diagonal (237, 240, 242, and 234) have strong values, indicating that the model has correctly classified most of the samples for the four classes. Misclassification is somehow trivial-an example being two Business articles predicted as Technology, and two World articles were misclassified into Business as well as Sports. The almost perfect classification of categories outlines very well the precision and recall of the model, with practically no confusion among the classes. The matrix exhibits reliability and hence robustness of the model in multi-class text classification.

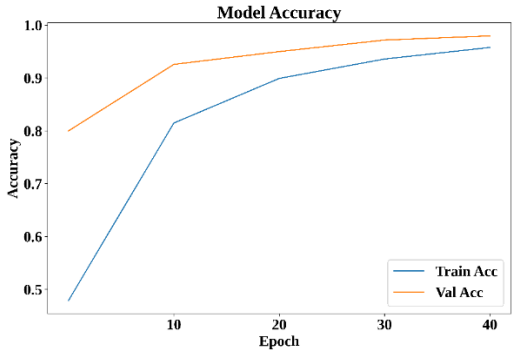

Figure 7: Overall accuracy graph.

Fig. 7 shows the training and validation accuracies of the model against 40 epochs. The blue line indicates the training accuracy, while the orange line shows validation accuracy. Initially, the training accuracy was near 48% and, after a sudden surge, shot past the 90% mark by the 30th epoch, continuing its slow ascent. The validation accuracy started relatively higher at around 80%, quickly shooting past 90% and eventually coming near 98%, thus denoting good generalization for the model with unseen data. As validation accuracy remains higher than training accuracy for most of the epochs, it signals no overfitting by the model and efficient learning. Both curves have a smooth upward trajectory, conveying that the model is well-trained and stable, with an excellent ability to predict.

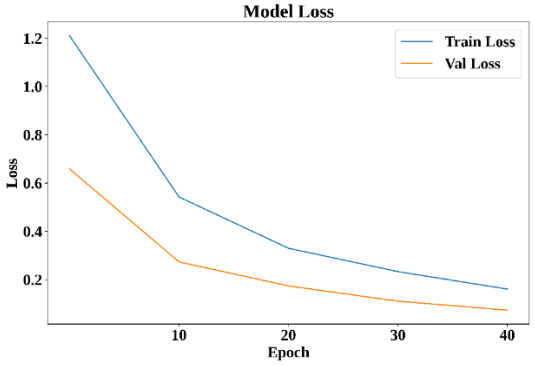

Figure 8: Overall loss graph.

Fig. 8 shows training and validation loss diminishing over 40 epochs, showcasing the model having learned. The training loss shown by the blue line starts at around 1.2 and steadily decreases to under 0.2, signaling the model's success in minimizing errors during training. The validation loss shown by the orange line drops from around 0.65 to almost 0.1, indicating steady improvement on unseen data. The validation loss staying lower than the training loss throughout is normally a sign that the model is generalizing well without overfitting. The smooth and downward slope of both curves demonstrates stable convergence and, thus, an effective training paradigm.

4.3 Impact of Training Configurations on Model Performance

Table 2 conducts a comparative study of the model performances under the various training settings, highlighting how architectural and hyperparameter settings impact the evaluation criteria. The "Original (Given)" setup is the latest and fully tuned proposed-version model, which achieves the best performance of all considered metrics.

Table 2: Performance Comparison Under Different Training Configurations.

Metric	Original (Given)	LR = 0.001	No Dropout	AdamW Optimizer

Accuracy	0.9797	0.9552	0.9211	0.9768
Precision	0.99	0.976	0.964	0.977
Recall	0.9795	0.9734	0.9695	0.9751
F1 Score	0.9796	0.9747	0.9717	0.976
fnr	0.0203	0.0266	0.0305	0.0249
fpr	0.0068	0.0084	0.0092	0.0076

Footnote: "Reported" refers to results directly taken from the original published papers, whereas "Re-implemented" refers to results reproduced in our experimental setup using the authors described configurations.

When the learning rate is raised to 0.001, we see a downgrade by a slight margin in performance in terms of accuracy (0.9797 to 0.9552) and F1 score (0.9747), indicating that a higher learning rate causes an unstable convergence. Yet there remains lesser degradation in performances when dropout is removed than when the learning rate is set at 0.001, dropping more in terms of Average Accuracy (0.9211) and F1 Score (0.9717), but increases the FNR and FPR. This study thus indicates that dropout helps in preventing overfitting.

Employing the AdamW optimizer, a decoupled weight decay version of Adam, produces a very slightly less than the original training setup accuracy of 0.9768 and F1 score of 0.976. These findings confirm that the original configuration, dropout, learning rate, and optimizer selected produce the cleanest and most robust performance.

5 Conclusion

This paper proposes a knowledge-aware multi-task paradigm to cope with few-shot-level, minimal supervision for text classification and entity-event extraction tasks. The model combines a joint BERT shared encoder with IA^3 adapters, spaCy semantic triple extraction, and TransE embeddings for external knowledge alignment. Following a common Bi-LSTM and softmax decoder for classification and extraction, the model learns simultaneously contextual semantic and structured information. This makes the model understand language better, which is also much more realistic for applications where label prediction and structured output would be produced side by side.

Based on an experimental evaluation, the model's efficacy was confirmed. Accordingly, it achieved an accuracy of 97.97%, a precision of 98.00%, the recall of 97.95%, and an F1 score of 97.96%, outperforming many other state-of-the-art methods. Ablation studies further emphasize the importance of knowledge integration and parameter-efficient tuning. Limitations of the current framework include evaluation on a single-domain benchmark without strict episodic few-shot settings, limited analysis of parameter efficiency, and partial validation of entity and event extraction performance. Subsequent work can address these gaps by assessing the model on true k-shot settings across multiple datasets, providing detailed efficiency comparisons with full fine-tuning approaches, and performing comprehensive evaluation on standardized extraction benchmarks to further strengthen empirical validation.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: MeiJie Zhao: Conceptualization, Methodology, Writing—Original Draft, Supervision. Yi Sun: Software, Formal Analysis, Visualization. Xuhao Fan: Data Curation, Investigation, Writing—Review & Editing. Yu Tian: Validation, Resources, Writing—Review & Editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset from AG News Classification used is publicly available for purposes of research. It can be accessed from the URL: https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset. No proprietary data has ever been used.

Ethics Approval: This study does not involve human participants, human data, or animals; therefore, ethical approval was not required.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schmidt T, Lange KR, Reccius M, Müller H, Roos M, Jentsch C. Identifying economic narratives in large text corpora—an integrated approach using Large Language Models. arXiv:2506.15041. 2025.
2. Fischer MT, Metz Y, Joos L, Miller M, Keim DA. MULTI-CASE: a transformer-based ethics-aware multimodal investigative intelligence framework. arXiv:2401.01955. 2024.
3. Sajikumar S. Emotion detection in text: a comprehensive analysis using classical, deep learning, and transformer-based models [Ph.D. thesis]. Dublin, Ireland: National College of Ireland; 2025.
4. Shi D, Zhang W, Yang J, Huang S, Chen X, Xu P, et al. A multimodal visual-language foundation model for computational ophthalmology. npj Digit Med. 2025;8(1):381. doi:10.1038/s41746-025-01772-2.
5. Sun Y, Wai Keung J, Kuen Yu H, Luo W. LogMeta: a few-shot model-agnostic meta-learning framework for robust and adaptive log anomaly detection. J Syst Softw. 2026;235:112781. doi:10.1016/j.jss.2026.112781.
6. Parovic M. Improving parameter-efficient cross-lingual transfer for low-resource languages [dissertation]. Cambridge, UK: University of Cambridge; 2024.
7. Prottasha NJ, Mahmud A, Sobuj MSI, Bhat P, Kowsher M, Yousefi N, et al. Parameter-efficient fine-tuning of large language models using semantic knowledge tuning. Sci Rep. 2024;14:30667. doi:10.1038/s41598-024-75599-4.
8. Harris S, Liu J, Hadi HJ, Ahmad N, Ali Alshara M. Benchmarking Hook and Bait Urdu news dataset for domain-agnostic and multilingual fake news detection using large language models. Sci Rep. 2025;15(1):15553. doi:10.1038/s41598-025-98271-x.
9. Xie X, Xu G, Zhao L, Guo R. OpenSearch-SQL: enhancing text-to-SQL with dynamic few-shot and consistency alignment. Proc ACM Manag Data. 2025;3(3):1–24. doi:10.1145/3725331.
10. Ahmad M. Toward a unified framework for information retrieval in large language model applications: balancing textual and graph-based knowledge sources [master's thesis]. Espoo, Finland: Aalto University; 2025 [cited 2026 Jan 1]. Available from: https://aaltodoc.aalto.fi/handle/123456789/136065.
11. Sankar NP, Manapure PS. EmotionSense: a deep learning-based text emotion classifier using NLP for real-time analysis. Preprint. 2025. https://doi.org/10.21203/rs.3.rs-6424907/v1.
12. Taha K. Smart: semantic, multi-objective, and reinforcement-based adversarial training for email spam detection. IEEE Access. 2025;13:112749–64.
13. Yin X, Fang W, Liu Z, Liu D. A novel multi-scale CNN and Bi-LSTM arbitration dense network model for low-rate DDoS attack detection. Sci Rep. 2024;14:5111. doi:10.1038/s41598-024-55814-y.
14. Creanga C, Dinu LP. Transformer based neural networks for emotion recognition in conversations. arXiv:2405.11222. 2024.
15. Li L, Sleem L, Gentile N, Nichil G, State R. Small language models in the real world: insights from industrial text classification. arXiv:2505.16078. 2025.
16. Liu H, Zhang F, Zhang X, Zhao S, Ma F, Wu XM, et al. Boosting few-shot text classification via distribution estimation. Proc AAAI Conf Artif Intell. 2023;37(11):13219–27. doi:10.1609/aaai.v37i11.26552.
17. Lei S, Zhang X, He J, Chen F, Lu CT. TART: improved few-shot text classification using task-adaptive reference transformation. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023 Jul 9–14; Toronto, Canada.
18. Wang Y, Li H, Teng F, Chen L. GORAG: graph-based online retrieval augmented generation for dynamic few-shot social media text classification. arXiv:2501.02844. 2025.
19. Hou G, Cao S, Ouyang D, Wang N. Label-template based few-shot text classification with contrastive learning. 2024: arXiv:2412.10110. 2024.
20. Lei T, Hu H, Luo Q, Peng D, Wang X. Adaptive meta-learner via gradient similarity for few-shot text classification. arXiv:2209.04702. 2022.
21. Luo D, Zhang C, Xu J, Wang B, Chen Y, Zhang Y, et al. Enhancing black-box few-shot text classification with prompt-based data augmentation. arXiv:2305.13785. 2023.
22. Kim HH, Woo D, Oh SJ, Cha JW, Han YS. ALP: data augmentation using lexicalized PCFGs for few-shot text classification. Proc AAAI Conf Artif Intell. 2022;36(10):10894–902. doi:10.1609/aaai.v36i10.21336.
23. Wang X, Du Y, Chen D, Li X, Chen X, Fan Y, et al. Improving domain-generalized few-shot text classification with multi-level distributional signatures. Appl Sci. 2023;13(2):1202. doi:10.3390/app13021202.
24. Li H, Huang G, Li Y, Zhang X, Wang Y, Li J. SEML: self-supervised information-enhanced meta-learning for few-shot text classification. Int J Comput Intell Syst. 2023;16(1):111. doi:10.1007/s44196-023-00287-6.

25. Wang J, Wang C, Luo F, Tan C, Qiu M, Yang F, et al. Towards unified prompt tuning for few-shot text classification. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022; 2022 Dec 7–11; Abu Dhabi, United Arab Emirates. p. 524–36. doi:10.18653/v1/2022.findings-emnlp.37.
26. Aljehani A, Hamid Hasan S, Ali Khan U. Advancing text classification: a systematic review of few-shot learning approaches. Int J Comput Digit Syst. 2025;17(1):1–21. doi:10.12785/ijcds/1571027526.
27. Anand A. AG news classification dataset. [cited 2026 Jan 1]. Available from: https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset.