Computing and Informatics, Vol. 44, 2025, 1445–1467, doi: 10.31577/cai_2025_6_1445

A LIGHTWEIGHT TEXT REPRESENTATION METHOD INTEGRATING TOPIC INFORMATION

Shuobin Zhang

College of Computer Science and Technology
Shandong Technology and Business University
Yantai, 264005, China
e-mail: zhangsb972@163.com

Jianhua Sun

Weihai Municipal Hospital, Cheeloo College of Medicine, Shandong University
Weihai, 264200, China
e-mail: 11259225@qq.com

Huanling Tang*, Wenhao Duan, Quansheng Dou

College of Computer Science and Technology
Shandong Technology and Business University
Co-Innovation Center of Shandong Colleges and Universities
Yantai, 264005, China
e-mail: {thl01, Dwh37818, li_dou}@163.com

Mingyu Lu

Information Science and Technology College
Dalian Maritime University
Dalian, 116026, China
e-mail: lumingyu@dlmu.edu.cn

* Corresponding author

Abstract. Transformer models and their variants have shown significant advantages in natural language processing tasks, but their high computational requirements limit their deployment on resource-constrained devices. To achieve the balance of computational cost and accuracy, the lightweight model pNLP-Mixer utilizes parameter-free projection to generate text embeddings. Nevertheless, the text embeddings generated by projection involve shallow semantic information and ignore the exploration of implicit semantic information. To overcome the challenges of high parameter cost and insufficient representation capabilities in existing models, we propose a method that incorporates topic information to enhance semantic richness. Our approach leverages the Latent Dirichlet Allocation (LDA) topic model to capture latent semantic relationships between words, thereby improving the expressiveness of text representations for downstream tasks. Building on this method, we propose a lightweight model named TEP-Mixer, which integrates multiple feature extraction modules to further enhance representation capabilities. Experimental results demonstrate that TEP-Mixer outperforms other lightweight models in accuracy while maintaining a lower parameter count across multiple benchmark datasets. It is suitable for resource-constrained devices.

Keywords: Lightweight, topic model, deep learning, text classification, feature representation

Mathematics Subject Classification 2010: 68T07

1 INTRODUCTION

Based on the Transformer architecture [1] and its variants, large pre-trained models have been widely used in complex language tasks, making them a mainstream tool in the field of natural language processing (NLP). Such tasks encompass text classification, natural language inference, question answering, and sentiment analysis. However, the computation of the self-attention mechanism requires a lot of computing resources. As a result, these large pre-trained models face considerable difficulties when deployed in resource-limited settings. Consequently, researchers are dedicated to achieving a balance between model size and predictive accuracy, aiming to extend the reach of artificial intelligence models to edge devices.

There are two principal strategies for mitigating the costs associated with model deployment. The first strategy encompasses model compression techniques, such as pruning, quantization, low-rank factorization, and knowledge distillation. The second strategy pertains to model reconstruction, which integrates lightweight modules to supplant the self-attention mechanism. It is noteworthy that the self-attention mechanism's influence on downstream tasks is not uniformly significant. Multilayer perceptrons (MLPs) have demonstrated their capacity to effectively replace the self-attention module, serving as the foundational network architecture, which has gained validation in the field of computer vision. Vision models like MLP-Mixer [2], ResMLP [3], Eff-CTM [4], CycleMLP [5], Sparse MLP [6], and MAXIM [7] have successfully leveraged MLPs across a range of image processing tasks, attaining commendable performance. Within the domain of NLP, models such as gMLP [8], pNLP-Mixer [9], HyperMixer [10], LCP-Mixer [11], TCA-Mixer [12], MHBA-Mixer [13], and TS-Mixer [14] have also realized significant achievements through the utilization of MLPs. Particularly, pNLP-Mixer abandons standard text representation methodologies. It incorporates a novel projection layer that eschews the need for training to produce text embeddings, substantially diminishing the model's parameters. Nonetheless, a considerable disparity persists between pNLP-Mixer and the cutting-edge language models. The projection layer of pNLP-Mixer may overlook the latent thematic aspects of the text, indicating that there is still room for improvement in the model's semantic representation capabilities.

Inspired by pNLP-Mixer and topic modeling, this paper introduces a lightweight text representation method and constructs a model named the Topic-Embedding Projection Mixer (TEP-Mixer). Similar to human information processing, TEP-Mixer retains the original surface-level text information while simultaneously extracting latent topic information using the Latent Dirichlet Allocation model [15]. This dual approach facilitates a more comprehensive understanding of the text's semantic content. By integrating topic information and leveraging prior knowledge, TEP-Mixer generates vector representations that encapsulate multi-level textual information, thereby enhancing the performance of subsequent NLP downstream tasks.

Through comprehensive evaluations, we demonstrate that the TEP-Mixer model achieves significant performance enhancements across multiple datasets, surpassing existing models such as gMLP, pNLP-Mixer, and HyperMixer. Notably, TEP-Mixer attains over 85 % of the performance of large-scale pre-trained models while maintaining a substantially smaller parameter size. This efficiency is particularly evident in the sequence labeling task on the MTOP dataset [16].

To further enhance adaptability, TEP-Mixer incorporates multiple feature extraction modules, including MLP-Mixer, LCP-Mixer, and MHBA-Mixer, allowing for dynamic selection based on specific task requirements. This flexibility significantly improves the model's generalization capabilities and robustness.

The main contributions of this paper are as follows:

- We propose a lightweight text representation method that integrates topic information, enhancing the semantic richness of text features.
- We design a flexible Feature-Mixer module that can dynamically select appropriate feature extraction techniques based on task requirements.
- We introduce the Topic-Embedding Projection Mixer (TEP-Mixer), which effectively combines projection-based methods with topic modeling.

- We validate the effectiveness of TEP-Mixer through extensive experiments on text classification and sequence labeling tasks, demonstrating significant performance improvements over existing models.

2 RELATED WORK

Since the introduction of pre-trained language models (PLMs) such as BERT [17] and GPT-4 [18], there has been a significant increase in the scale of NLP models. These expansive models use large datasets for unsupervised pre-training through autoencoding or autoregressive techniques, adeptly capturing nuanced semantic details from text and leading to substantial improvements in performance across a variety of NLP tasks. However, due to their reliance on multi-head attention or self-attention mechanisms, PLMs often have billions or even tens of billions of parameters, which require considerable computational resources. This makes their deployment on resource-constrained devices, like mobile devices or embedded systems, particularly challenging. Therefore, achieving an optimal balance between model complexity and accuracy has become a key issue in both academic research and industrial applications.

To address the delicate balance between model performance and computational cost, researchers have developed a variety of model compression techniques, including pruning, quantization, knowledge distillation, and low-rank decomposition. These methods aim to reduce model complexity in different ways, focusing on reducing the number of parameters, computational load, and memory footprint. Notably, knowledge distillation allows the original model to incorporate external knowledge, enriching the semantic depth of text representations and producing more detailed features. For example, TinyBERT [19] and MobileBERT [20] use knowledge distillation to distill the insights of BERT into more compact models. Despite these models being smaller, they still have considerable parameter sizes, which can make their deployment on resource-constrained devices challenging. Recognizing that compressed models still rely on large word embedding matrices, recent research has investigated methods like locality-sensitive hashing to generate word embeddings, reducing the models' dependency on embedding matrices. Models like PRADO [21] and pQRNN [22] use projection-based methods to create word embedding representations, effectively reducing the number of vocabulary parameters and significantly reducing the model's overall size. These innovations broaden the horizons for practical NLP applications, providing a wider range of options for deploying NLP technology.

An alternative strategy involves employing a multi-layer perceptron (MLPs) to replace the self-attention mechanism. MLPs have demonstrated promising results in both computer vision and natural language processing. In computer vision, MLP-Mixer, which comprises a token mixing layer and a channel mixing layer, relies solely on fundamental matrix operations and non-linear transformations to underscore the MLP's considerable potential. In natural language processing, models such as gMLP, pNLP-Mixer, and HyperMixer have demonstrated comparable advancements. Notably, pNLP-Mixer stands out as the first ultra-compact model in NLP to feature a full MLP architecture. It integrates projection-based techniques with a lightweight MLP framework, incorporating a novel projection layer that eliminates the need for training, bypassing the traditional embedding layer. This layer directly projects words into a lower-dimensional vector space, thereby precluding the model from learning embeddings during training and capping the model's parameter count at 1 million. Figure 1 provides a schematic representation of the pNLP-Mixer architecture.

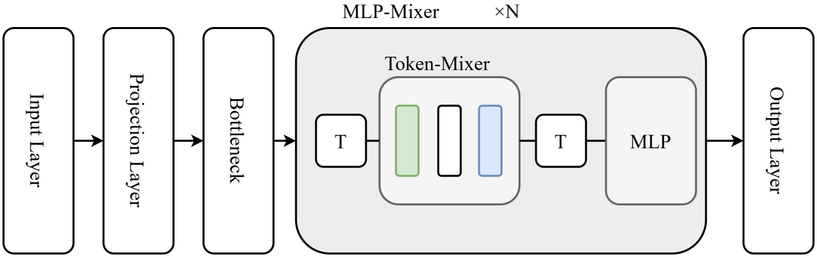


Figure 1. Overall architecture of pNLP-Mixer

However, the projection layer in pNLP-Mixer might not fully capture the subtle semantic variations present in text, which could result in less than optimal representation capabilities. Considering that textual content encompasses more than just the surface-level semantics of individual words and extends to higher-level semantic dimensions, such as implicit topic information, this paper uses the incorporation of prior knowledge to enhance text representation. Utilizing topic modeling, which effectively identifies underlying thematic structures within text, this method aims to capture the nuanced semantic details embedding within textual content. Consequently, this paper introduces an unsupervised topic model to delineate the topic structure throughout textual data. Employing this topic structure as pre-existing knowledge, the paper enriches the semantic content of texts, leading to an enhancement in the execution of subsequent NLP tasks.

3 METHODS

In natural language processing, large-scale pre-trained models often have a substantial number of parameters, making their deployment impractical for resource-constrained environments. Conversely, while lightweight models are efficient in terms of parameters, they often lack sufficient representational power, which restricts their ability to perform well on complex tasks. To address these dual challenges, this paper proposes a lightweight text representation method that integrates topic information, introducing a model named TEP-Mixer. TEP-Mixer aims to achieve a balance between parameter efficiency and strong representation ability. In this section, we present the overall architecture of TEP-Mixer and provide a detailed explanation of the Topic-Embedding Projection feature representation method.

3.1 Overall Architecure

The model architecture is illustrated in Figure 2. Firstly, the TEP-Mixer model preprocesses the text data and converts it into vector representations using the Topic-Embedding Projection layer, which comprises the Projection layer, the Topic-Embedding layer, and a Fusion layer. Sequence information in the key features is then captured through a selectable feature extraction layer called the Feature-Mixer. Finally, a prediction layer performs classification.

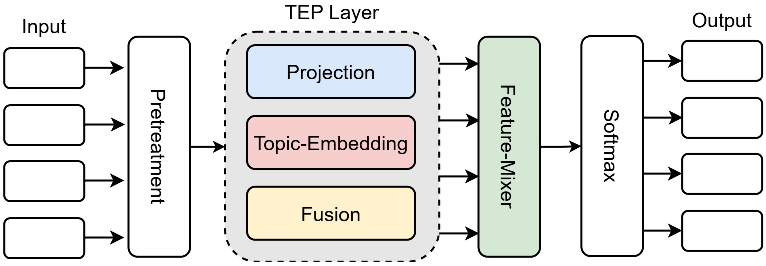


Figure 2. Overall architecture of TEP-Mixer

By incorporating topic information alongside traditional semantic features, the model better understands the overall context and deeper topic structures within the text. This allows the text features to capture not only basic lexical semantics but also higher-level semantic relationships, resulting in semantically rich representations of text features. Consequently, the model can make more accurate judgments in subsequent NLP tasks.

3.2 Topic-Embedding Projection Layer

The Topic-Embedding Projection (TEP) layer is composed of three main components: the Projection layer, the Topic-Embedding layer, and the Fusion layer. The Topic-Embedding layer leverages a Latent Dirichlet Allocation (LDA) model to generate topic distributions for each token, effectively capturing the latent semantic information inherent in the tokens. This approach enhances traditional projection techniques by providing a more holistic and semantically enriched feature representation.

The detailed architecture of the TEP layer is illustrated in Figure 3. This design offers a robust alternative for text representation, improving model performance while maintaining computational efficiency.

Formally, the input features are represented as $\mathbf{X} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$, where i denotes the position of each token within the document. As depicted in Fig-

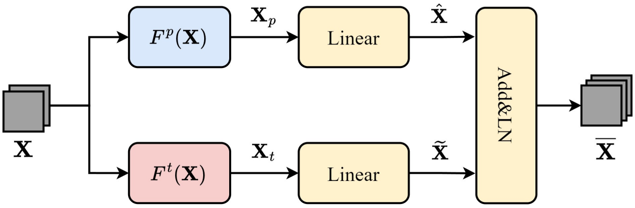


Figure 3. Specific implementation of TEP layer

ure 3, the preprocessed input sequence $\mathbf{X}$ undergoes sequential processing through the Projection layer and the Topic-Embedding layer. The resulting representations are then integrated in the Fusion layer, which combines the projected features with the topic embeddings to produce a unified feature representation.

3.2.1 Projection Layer

The Projection Layer captures morphological features by transforming each token into a compact representation. Following the approach used in pNLP-Mixer [9], each subword in the vocabulary V is mapped to a set of 64 hash codes. Among these, the smallest hash code is selected as the minimum hash value, representing the token in the projected feature space. This transformation can be mathematically defined as:

$$\mathbf{X}_p = F^p(\mathbf{X}), \tag{1}$$

where $\mathbf{X}$ is the input text sequence, $\mathbf{X}_p$ is the projected feature vector in $\mathbb{R}^{d_p}$, and d_p denotes the dimensionality of this projection feature.

3.2.2 Topic-Embedding Layer

LDA is a generative probability model, and its training procedure is shown in Algorithm 1.

To embed semantic depth, the Topic-Embedding Layer leverages the LDA topic model. LDA assigns each token in the vocabulary V a probability distribution across various topics, effectively capturing the thematic distribution within the text. This layer provides additional context by enriching tokens with topic distributions that reflect semantic content. As LDA operates independently of the training process, it encodes the topic distribution based on pre-computed token identifiers, which reduces inference time and enhances reusability across models. The topic embedding can be represented as:

$$\mathbf{X}_t = F^t(\mathbf{X}) = \phi_t, \tag{2}$$

where $\mathbf{X}$ is the input text sequence, $\mathbf{X}_t$ is the projected feature vector in $\mathbb{R}^{d_t}$, and d_t denotes the dimensionality of this topic embedding.

Algorithm 1 LDA Model Training Algorithm (Gibbs Sampling)

```
Require:
Training set $D = \{d_1, d_2, \ldots, d_M\}$, where $M$ is the number of documents
Number of topics $K$
Number of iterations $T$
Hyperparameters $\alpha$ and $\beta$ for Dirichlet distribution
Ensure:
Word-topic distribution $\phi$

for $m = 1$ to $M$ do
    for each word $w$ in document $m$ do
        Randomly assign a topic $z_w$ to word $w$
        Update document-topic count $n_{mk}$ and topic-word count $n_{kw}$
    end for
end for
for $t = 1$ to $T$ do
    for $m = 1$ to $M$ do
        for each word $w$ in document $m$ do
            Record current topic assignment $z_w^{(m)}$
            Decrease counts $n_{mk}$ and $n_{kw}$ for the current topic
            Compute conditional probabilities for reassigning $w$ to other topics:
                $p(z_w = k \mid z_{-w}, w, \alpha, \beta) \propto (n_{mk}^{-w} + \alpha) \frac{n_{kw}^{-w} + \beta}{n_k^{-w} + W\beta}$
            Resample a new topic for word $w$ from the conditional probabilities
            Update $n_{mk}$ and $n_{kw}$ for the new topic
        end for
    end for
end for
For each word $w$, calculate $\phi_w$
return $\phi$
```

3.2.3 Fusion Layer

The Fusion Layer combines the projection and topic features to create a unified representation. As shown in Figure 3, we first apply linear transformations and non-linear activation functions to both the projected and topic-embedded features, capturing higher-level interactions. This process is illustrated as Equation (3):

$$
\begin{aligned}
\hat{\mathbf{X}} &= \sigma(W_1 \mathbf{X}_p + b_1), \\
\widetilde{\mathbf{X}} &= \sigma(W_2 \mathbf{X}_t + b_2),
\end{aligned}
\tag{3}
$$

where $\sigma(\cdot)$ is the activation function (e.g., GELU), and W_1, W_2, b_1, and b_2 are trainable parameters in the fully connected layers. To maintain a lightweight architecture, TEP uses a straightforward addition and Layer Normalization (LN) to merge the transformed features as Equation (4):

$$\overline{\mathbf{X}} = LN\left(\hat{\mathbf{X}} + \widetilde{\mathbf{X}}\right), \tag{4}$$

where $\overline{\mathbf{X}}$ serves as the input for the subsequent Feature-Mixer layer.This fusion mechanism effectively integrates morphological projection features with thematic topic features, allowing the model to capture both surface-level and thematic information.

The TEP layer seamlessly integrates the strengths of projection layers with topic modeling, leveraging LDA to extract thematic information from individual words. Operating independently of the training process, it encodes topic distributions based on token identifiers, enriching the feature representation with a pre-computed topic distribution table for tokens and imbuing the text representation with deeper semantic context.

3.3 Feature-Mixer

To adapt to different data and task requirements, we designed a replaceable feature mixer module called Feature-Mixer. The module provides a variety of feature extraction methods including MLP-Mixer, LCP-Mixer [11], and MHBA-Mixer [13]. Among them, LCP-Mixer and MHBA-Mixer have been proposed in our previous lightweight research.

3.3.1 MLP-Mixer

MLP-Mixer [2] is composed of multiple stacked modules, each containing two Multilayer perceptrons (MLPs) and several transpose operations. Within this framework, the initial MLP imposes a nonlinear transformation on each channel of the input data. Following this, a transpose operation directs the output of each position to the subsequent MLP, thereby enabling the exchange of information across channels and positions. This paper employs the original design of MLP-Mixer, utilizing input data sourced from the TEP layer. This input is represented as a matrix $\overline{\mathbf{X}} \in \mathbb{R}^{n \times h}$, where n signifies the text's sequence length, and h denotes the embedding dimension. Subsequently, $\overline{\mathbf{X}}$ is processed by MLP-Mixer to produce the output $\mathbf{O} \in \mathbb{R}^{n \times h}$.

3.3.2 LCP-Mixer

LCP-Mixer [11] is shown in Figure 4. It consists of a direct mapping module, a local information perception module, and a global information perception module. In the local information perception module, the text is divided into different groups and blocks based on the concept hierarchy to perceive local information at different

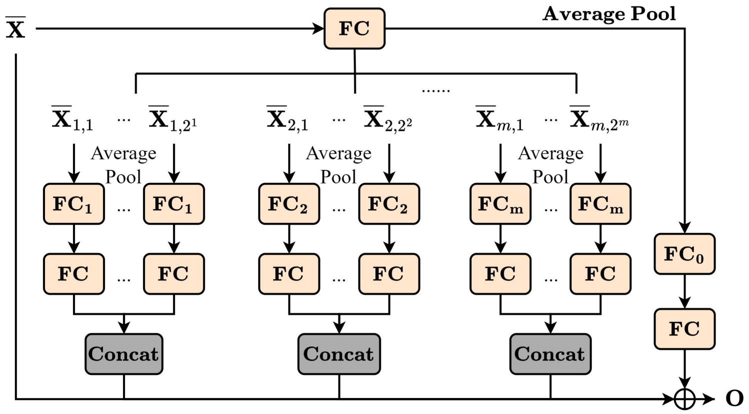

$\overline{\mathbf{X}}$
FC
Average Pool
......
$\overline{\mathbf{X}}_{1,1}$ ... $\overline{\mathbf{X}}_{1,2^1}$
$\overline{\mathbf{X}}_{2,1}$ ... $\overline{\mathbf{X}}_{2,2^2}$
$\overline{\mathbf{X}}_{m,1}$ ... $\overline{\mathbf{X}}_{m,2^m}$
Average Pool
Average Pool
Average Pool
$\mathbf{FC_1}$ ... $\mathbf{FC_1}$
$\mathbf{FC_2}$ ... $\mathbf{FC_2}$
$\mathbf{FC_m}$ ... $\mathbf{FC_m}$
FC ... FC
FC ... FC
FC ... FC
Concat
Concat
Concat
$\mathbf{FC_0}$
FC
O

Figure 4. LCP-Mixer

levels. The average pooling operation AvgPool is used to aggregate the information within each block to obtain the local feature representation $\overline{\mathbf{X}}_{m,k}$.

Local feature $\overline{\mathbf{X}}_{\text{loc}}$ is obtained by concatenating features $\overline{\mathbf{X}}_{m,k}$ from different groups. In the global information perception module, the nonlinear features $\overline{\mathbf{X}}$ are average pooled to obtain the global token representation $\overline{\mathbf{X}}_{\text{global}}$. The local features and the global features are remapped to the original feature space through the linear mapping layer and the channel projection layer FC. Finally, the original features $\overline{\mathbf{X}}$, local features $\overline{\mathbf{X}}_{\text{loc}}$, and global features $\overline{\mathbf{X}}_{\text{global}}$ are added together to obtain the final output features $\mathbf{O}$.

3.3.3 MHBA-Mixer

The Multi-Head Hidden Bias Attention (MHBA) [13] is depicted in Figure 5. In this methodology, tokens encapsulate both projection and topic information. Initially, the input tokens matrix is segmented into h sub-matrices. Convolutional layers are then independently applied to each sub-matrix to deduce the profound semantic attributes of the tokens, yielding the representation $\overline{\mathbf{X}}^i_{\text{at}}$, where i iterates from 1 to h, representing each head in the multi-head attention mechanism.

Subsequently, a binomial sampling layer is employed to sparsify the tokens, generating a sparse representation matrix $\overline{\mathbf{X}}^i_w$. This not only directs the model towards salient information but also alleviates computational overhead. The hybrid information between the sparse representation matrix and the original sub-matrix is then computed to augment the semantic articulation capability of the tokens, resulting in the feature $\overline{\mathbf{X}}^i_{\text{dt}}$.

Moreover, global information $\overline{\mathbf{X}}^i_{\text{ag}}$ is extracted from each sub-matrix to grasp the contextual interrelations across the entire sequence. This global information is interactively combined with $\overline{\mathbf{X}}^i_{\text{dt}}$ to formulate the attention output for each head,

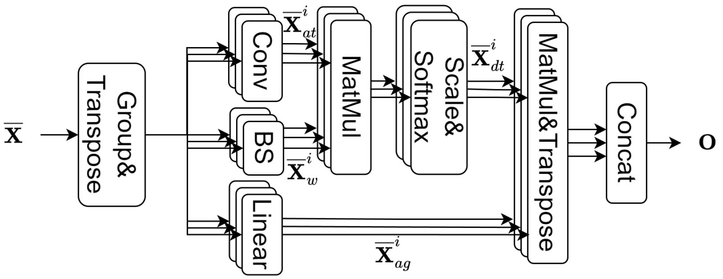

$\overline{\mathbf{X}}$
Group& Transpose
Conv
BS
Linear
$\overline{\mathbf{X}}^i_{at}$
$\overline{\mathbf{X}}^i_{w}$
$\overline{\mathbf{X}}^i_{ag}$
MatMul
Scale& Softmax
$\overline{\mathbf{X}}^i_{dt}$
MatMul&Transpose
Concat
$\mathbf{O}$

Figure 5. Multi-head HBA consists of several attention layers running in parallel

denoted as head$^i(\overline{\mathbf{X}})$. It is crucial to note that this process is conducted for each head i, where i ranges from 1 to h. Ultimately, the attention outputs head$^i(\overline{\mathbf{X}})$ across all heads are concatenated to procure a holistic attention representation $\mathbf{O}$. This approach effectively amalgamates local features with global context, thereby enhancing the model's representational prowess and contextual comprehension through the concatenation of multi-head attention outputs.

3.4 Prediction Layer

For the text classification tasks, a softmax layer is integrated subsequent to the Feature-Mixer's output to convert it into a probability distribution across categories. The loss objective function for the TEP-Mixer is delineated in Equation (5):

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{C} y_i \log(\hat{y}_i) + \lambda \cdot \Omega(\theta), \tag{5}$$

where $\mathbf{y}$ represents the actual labels, $\hat{\mathbf{y}}$ represents the predicted probability distribution, N is the number of samples, C is the number of categories, λ is the regularization weight, and $\Omega(\theta)$ refers to the L2 regularization term.

In the sequence labeling tasks, a linear layer is appended following the output of Feature-Mixer to transform the output into a categorical probability distribution. This task also uses the cross entropy loss as shown in Equation (6):

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N \cdot T}\sum_{j=1}^{N}\sum_{t=1}^{T}\sum_{c=1}^{C} y_{j,t,c} \log(\hat{y}_{j,t,c}) + \lambda \cdot \Omega(\theta), \tag{6}$$

where $\mathbf{y}$ signifies the actual sequence of labels, $\hat{\mathbf{y}}$ represents the model's predicted probability distribution for the labels, N denotes the size of the training set, T signifies the length of the sequence, which corresponds to the count of tokens present in each sample, j represents the j^{th} sample, t represents the t^{th} position of the sequence, c represents the c^{th} category, and C is the total number of categories. The regularization weight is represented by λ, and $\Omega(\theta)$ refers to the L2 regularization term.

Optimizing the model parameters in accordance with the objective function enables the effective integration of the proposed topic-enriched text representation with the neural network, thereby enhancing its capability to address NLP downstream tasks.

4 EXPERIMENT

This paper conducts experiments on text classification and sequence labeling tasks, and the results show that lightweight text representation methods that incorporate topic information can achieve competitive performance in lower-dimensional neural network models. This method is primarily chosen for its ability to incorporate prior knowledge, thereby providing high-quality input features for subsequent models. The effectiveness and generalization capability of the proposed lightweight text representation method, which integrates thematic information, are discussed in the following sections. Table 1 presents the utilized dataset, while Table 2 illustrates the relevant comparison models.

4.1 Experimental Setup

This paper leverages well-established public datasets for experimental validation.

The IMDb[1] dataset, derived from the Internet Movie Database, consists of movie reviews and is a standard resource for sentiment classification on extensive textual data. It features user-generated reviews labeled with sentiment polarities, indicating positive or negative opinions of the movies.

The SST-2[2] dataset, extracted from movie review platforms, is dedicated to emotion classification. It includes texts annotated with emotional polarities—positive, negative, and neutral. With approximately 67 000 sentences divided into training, validation, and test subsets, the SST-2 is used for model training, tuning, and evaluation in sentiment analysis.

The AGNews[3] dataset aggregates internet news articles for news topic classification. It encompasses articles sorted into four categories: Sports, Technology, Entertainment, and Business. Each entry comprises a title and description, totaling 1.2 million training instances and 7 600 test instances. This dataset serves as a benchmark for assessing text classification models.

The CoLA dataset is crafted for sentence acceptability judgment tasks. It contains sentences to be judged for grammatical acceptability, sourced from linguistic publications. The CoLA comprises 8 550 training samples and 1 063 test samples,

[1] https://huggingface.co/datasets/stanfordnlp/imdb

[2] https://huggingface.co/datasets/nyu-mll/glue

[3] https://huggingface.co/datasets/fancyzhx/ag_news

and is a standard tool for gauging the efficacy of language models and parsing systems.

The MTOP[4] dataset is a multilingual task-oriented semantic parsing dataset covering 6 languages and 11 domains. It comprises 11 500 training samples and 3 000 test samples.

Detailed information is presented in Table 1.

Tasks	Dataset	Classes	Length	Training	Test
Semantic Analysis	IMDb	2	1 024	25 K	25 K
	SST-2	2	64	67 K	1.8 K
Text Categorization	AGnews	4	128	120 K	7.6 K
Natural Language Inference	CoLA	2	32	8.55 K	1.06 K
Sequence Labeling	MTOP	78	64	11.5 K	3 K

Table 1. Dataset information

Evaluation Metrics: This paper evaluates model efficacy and scalability by quantifying model size in terms of parameters and primarily assessing model performance using accuracy. The trade-off between model complexity and performance is scrutinized through an analysis of these metrics.

Training Setup: The model training is conducted on a system with an RTX4060 GPU, which has 8 GB memory and is complemented by 64 GB of RAM. The implementation is carried out using the PyTorch Lightning extension version 2.0.5.

4.2 Compare Models

Table 2 provides a comparative analysis of various models, with a focus on those that have achieved the state-of-the-art performance. The models pNLP-Mixer, Hypermixer, and TinyBert are evaluated based on their performance metrics. In contrast, RoBERTa [23], XLNet [24], BERT Large, UDA, BERT-ITPT-FiT [25], gMLP, Longformer [26], FNet [27], and MobileBert are assessed based on their parameter count.

It should be noted that because different datasets have different maximum sentence lengths, the number of parameters for pNLP-Mixer and TEP-Mixer is not fixed.

4.3 Ablation Experiments

This section presents a thorough ablation study aimed at evaluating the effectiveness and generalizability of the lightweight text representation methods that incorporate topic information. It follows with an in-depth analysis and discussion.

[4] `https://huggingface.co/datasets/iohadrubin/mtop`

Model	Pre-trained	Architecture	Parameters (M)
RoBERTa	✓	Attention	125
XLNet	✓	Attention	240
Bert Large	✓	Attention	340
UDA	✓	Attention	340
Bert-ITPT-FiT	✓	Attention	340
Longformer	✓	Attention	149
FNet	✓	Attention	85
MobileBert	✓	Attention	25.3
TinyBert	✓	Attention	14.5
gMLP	✓	MLP	365
HyperMixer	×	MLP	11
pNLP-Mixer	×	MLP	1.2-2.1
TEP-Mixer	×	MLP	0.7-1.7

Table 2. The information of compared models with TEP-Mixer

4.3.1 Hyperparameter Settings

To ensure a fair comparison, the learning rate was uniformly set to $5e^{-4}$, and the hidden dimension was consistently maintained at 256 for the duration of the experiment. This method ensures consistency across all trials. Furthermore, a weight regularization term, specifically L2 regularization, was applied with a value of $1e^{-8}$ to reduce the likelihood of model overfitting. Concurrently, the number of topics was fixed at 64, and the hyperparameters α and β were assigned the values of $4e^{-2}$ and $1e^{-2}$.

4.3.2 The Impact of Each Module on the Performance of TEP-Mixer

This paper delves into the impact of various components by conducting experimental evaluations on the SST2, AGnews, and IMDb datasets. Specifically, the paper assesses the following modifications:

1. the removal of the multi-layer perceptron mixing layer;
2. the removal of the projection layer; and
3. the removal of the topic embedding layer.

Model	LDA	Projection	MLP	AGnews	SST2	IMDb	Params (M)
TEP-Mixer	✓	✓	✓	92.36	82.91	87.16	0.7–1.7
w/o LDA	×	✓	✓	91.13	80.69	79.67	0.7–1.7
w/o Projection	✓	×	✓	90.79	72.47	87.07	0.15
w/o MLP	✓	✓	×	92.46	80.84	88.71	0.5–1.5

Table 3. The impact of different modules on the model

Table 3 demonstrates that the reliance on various modules of TEP-Mixer differs across datasets. For example, the removal of the projection layer causes a notable decrease in performance for the AGnews and SST2 datasets, with drops of 1.57 % and 10.44 %, respectively. This suggests a high sensitivity to the information derived from projection. In contrast, the elimination of the multi-layer perceptron (MLP) mixing layer is associated with performance gains for the AGnews and IMDb datasets, increasing to 92.46 % and 88.71 %, respectively. This indicates a favorable interaction with the MLP and a lower necessity for extensive information. Additionally, the removal of the topic embedding layer causes a decline in performance across all datasets, highlighting the essential contribution of the embedding topic knowledge to the model's efficacy.

4.3.3 Comparison of Different Topic Numbers

To examine the impact of varying the number of topics on the TEP-Mixer, a series of experiments were conducted across the SST2, AGnews, IMDb and MTOP datasets. The experiments varied the number of topics from 32 to 80, incrementing by 8, while all other hyperparameters were held constant.

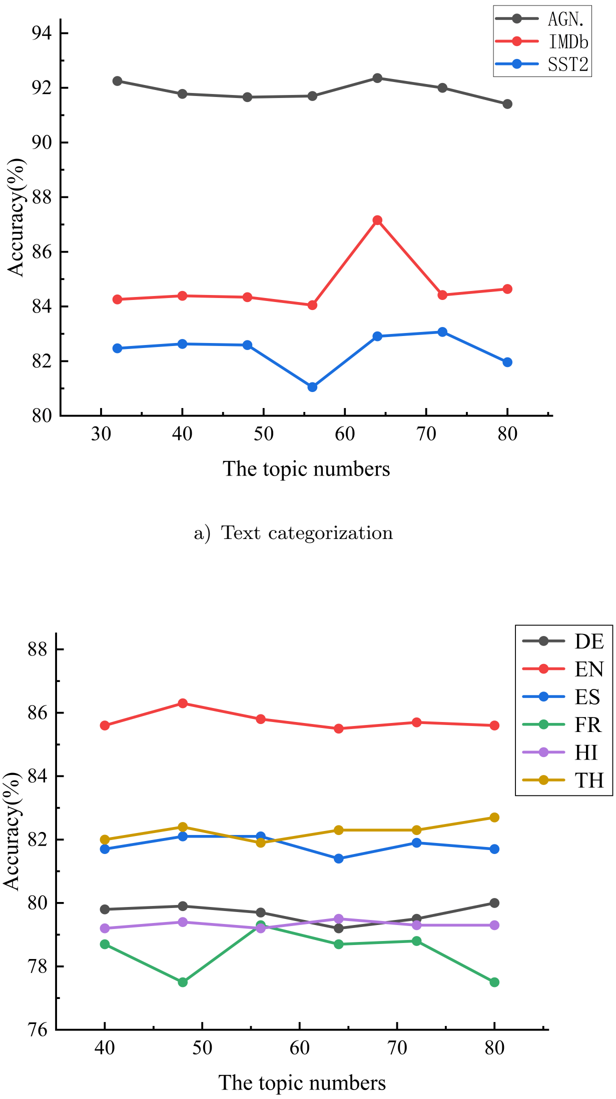
Figure 6 demonstrates that different datasets have varying sensitivities to the number of topics, yet the general trend is consistent. The model attains peak performance in text classification tasks when the topic count is 64. Concurrently, for sequence labeling tasks, optimal performance is attained at 56 topics. Although augmenting the number of topics can improve performance, it concurrently results in a growth of model parameters, escalating the demand for computational resources. After weighing the trade-off between computational resources and performance, the decision was made to set the topic count at 64 for text classification and at 56 for sequence labeling. This choice ensures a model complexity adequate for learning data characteristics without incurring high computational costs.

4.3.4 Architectural Exploration

When exploring the model architecture, we compared the performance of three different Feature-Mixer modules, MLP-Mixer, LCP-Mixer and MHBA-Mixer, on text classification and sequence labeling tasks.

As can be seen from Tables 4 and 5, it shows that on text classification tasks, the performance of the models has improved after introducing the TEP layer. In particular, MHBA-Mixer, with the assistance of the TEP layer, achieved the highest accuracy of 89.45 % on the IMDb dataset, and had an advantage in the number of parameters, only 0.7 M, showing good lightweight properties.

However, on the sequence labeling task, the situation is different. MLP-Mixer outperforms other models in all languages, especially with the support of the TEP layer, its average accuracy reaches 81.1 %, which is significantly higher than other models. This shows that in tasks such as sequence labeling, the structure of MLP-Mixer may be more suitable for processing the dependencies of sequence data.



a) Text categorization

b) Sequence labeling

Figure 6. The impact of topic number

Model	Accuracy (%)			Param (M)
	AGnews	SST2	IMDb	
pNLP-Mixer	90.00	78.89	78.60	1.2–2.1
TEP-Mixer(MLP)	92.36	83.26	87.16	0.7–1.7
LCP-Mixer	91.85	83.41	88.20	0.2
TEP-Mixer(LCP)	92.31	82.53	**89.45**	**0.2**
MHBA-Mixer	91.73	83.48	87.88	0.7
TEP-Mixer(MHBA)	**92.61**	**83.57**	88.52	0.7

Table 4. Comparison of different architectures using the TEP layer on the validation set of AGnews, SST2 and IMDb

Model	Param (M)	Accuracy (%)						AVG(%)
		EN	ES	FR	DE	HI	TH	
pNLP-Mixer	1 (8 bit)	84.0	78.3	75.2	76.9	76.5	74.1	77.5
LCP-Mixer	0.2 (8 bit)	79.2	77.2	73.9	74.6	75.0	75.9	76.0
TEP-Mixer(MLP)	0.7 (8 bit)	**85.5**	**81.4**	**78.7**	**79.2**	**79.5**	**82.3**	**81.1**
TEP-Mixer(LCP)	1 (8 bit)	81.4	77.2	76.5	76.9	76.0	78.4	77.7
TEP-Mixer(MHBA)	0.7 (8 bit)	67.2	63.5	61.4	65.9	61.2	61.3	63.4

Table 5. Comparison of different architectures using the TEP layer on the validation set of MTOP

Overall, no model is perfect for all tasks. In text classification tasks, MHBA-Mixer can be the first choice for feature extraction modules due to its small number of parameters and excellent performance. In the sequence labeling task, MLP-Mixer becomes a more suitable choice due to its stable and high-performance performance in multiple languages.

4.4 Results

4.4.1 Text Classification

This paper evaluates the generalizability of the TEP-Mixer through experimental assessments on a variety of widely utilized public datasets, with the findings detailed in Table 6.

In terms of parameter count, the TEP-Mixer is significantly more parameter-efficient. When compared to expansive pre-trained models such as XLNet, BERT Large, UDA, Bert-ITPT-FiT, gMLP, and others, TEP-Mixer boasts approximately an order of magnitude fewer parameters. Despite this, its performance on the AGnews dataset is on par with that of XLNet and Bert-ITPT-FiT, with performance deficits of merely 3.4 % and 3 %, respectively. Furthermore, on the IMDb dataset, TEP-Mixer not only outperforms TinyBERT but also matches 90.59 % of XLNet's performance. Remarkably, TEP-Mixer attains an accuracy of 70.64 % on the CoLA dataset, outstripping some of the larger pre-trained models. In comparison with

Model	Accuracy (%)				Param (M)
	AGnews	SST2	CoLA	IMDb	
RoBERTa	/	**96.70**	67.80	95.30	125
XLNet	**95.55**	94.40	69.00	**96.21**	240
Bert Large	/	93.70	/	95.49	340
UDA	/	/	/	95.80	340
Bert-ITPT-FiT	95.20	/	/	/	340
gMLP	/	94.80	/	/	365
Longformer	/	/	/	95.70	149
FNet	/	94.00	67.00	/	85
TinyBert	/	92.60	43.30	71.30	14.5
MobileBERT	/	92.80	51.10	/	25.3
HyperMixer	/	80.70	/	/	12.5
pNLP-Mixer	90.00	78.89	69.69	78.60	1.2–2.1
TEP-Mixer (ours)	92.36	83.26	**70.64**	87.16	**0.7–1.7**

Table 6. Accuracy on different test sets in text classification tasks. Among them, Param (M) represents the parameter quantity, and '/' represents no available data.

models of similar scale, this paper discovered that TEP-Mixer shows improvements of 2.36 % on AGnews, 4.8 % on SST2, 1.3 % on CoLA, and 9.8 % on IMDb when compared to pNLP-Mixer.

4.4.2 Sequence Labeling

This paper evaluates the efficacy of TEP-Mixer on the sequence labeling task, utilizing the MTOP dataset, and benchmarks it against several models. The outcomes are detailed in Table 7. To ensure equitable comparison, this paper standardizes the TEP-Mixer's metrics against those of pQRNN and pNLP-Mixer, with performance values for all baselines sourced from pNLP-Mixer experiments. Additionally, TEP-Mixer is quantized on the 8-bit version.

Table 7 indicates that the TEP-Mixer has a comparable parameter count to that of pNLP-Mixer. It exhibits slightly lower performance than XLUM-R and mBERT on the Spanish and French test sets (ES, FR). Nonetheless, TEP-Mixer achieves superior performance on all other datasets. Strikingly, TEP-Mixer's performance is markedly higher while its parameter count is substantially lower than that of the other models under comparison.

4.4.3 Discussion

Overall, while TEP-Mixer may not yet match the state-of-the-art (SOTA) models in all respects, it offers a notable advantage in terms of parameter efficiency. When juxtaposed with analogous models, TEP-Mixer demonstrates commendable performance across both parameter scale and model efficacy. For instance, on the sequence labeling task, TEP-Mixer sustains a lower parameter count on the MTOP

Model	Params. (M)	Accuracy (%)						AVG(%)
		EN	ES	FR	DE	HI	TH	
XLU	70 (float)	78.2	70.8	68.9	65.1	62.6	68.0	68.9
XLUM-R	550 (float)	85.3	81.6	79.4	76.9	76.8	73.8	79.0
mBERT	170 (float)	84.4	**81.8**	**79.7**	76.5	73.8	72.0	78.0
Transformer	2 (float)	71.7	68.2	65.1	64.1	59.1	48.4	62.8
pQRNN	2 (8 bit)	78.8	75.1	71.9	68.2	69.3	68.4	71.9
pQRNN-distilled	2 (8 bit)	79.4	75.4	73.0	68.6	70.2	69.5	72.7
pNLP-Mixer	1 (8 bit)	84.0	78.3	75.2	76.9	76.5	74.1	77.5
TEP-Mixer (ours)	0.7 (8 bit)	**85.5**	81.4	78.7	**79.2**	**79.5**	**82.3**	**81.1**

Table 7. Cross-language matching accuracy on the MTOP test set. For each column, we highlight the best overall performance and mark the model with the best performance in bold.

dataset, yet it surpasses the performance of several large pre-trained models. This underscores TEP-Mixer's competitiveness as a compact model. Concurrently, the topical information embedding within texts, when integrated as prior knowledge into various neural network architectures, can exert a beneficial influence, particularly on datasets with heightened sensitivity to topical nuances.

5 CONCLUSION

In conclusion, this paper presents a lightweight text representation method that integrates topic information, culminating in the development of the TEP-Mixer model. The model utilizes the LDA topic model to uncover the thematic distributions of words and integrates this prior knowledge into the projection layer. This integration allows the model to better capture diverse types of information. To assess the model's generalization ability, we compared different feature extraction modules across multiple datasets. Experimental results indicate that the proposed topic-enhanced text representation improves the performance of neural network models in several downstream NLP tasks, particularly in resource-constrained environments. Although TEP-Mixer still falls short compared to state-of-the-art models, it aims to strike an effective balance between performance and computational efficiency. Future research will focus on incorporating prior knowledge from traditional machine learning algorithms to further enhance model performance and interpretability.

REFERENCES

[1] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, Ł.—Polosukhin, I.: Attention Is All You Need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S.,

Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 5998–6008, doi: 10.48550/arXiv.1706.03762.

[2] Tolstikhin, I. O.—Houlsby, N.—Kolesnikov, A.—Beyer, L.—Zhai, X.—Unterthiner, T.—Yung, J.—Steiner, A.—Keysers, D.—Uszkoreit, J.—Lucic, M.—Dosovitskiy, A.: MLP-Mixer: An All-MLP Architecture for Vision. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., Wortman Vaughan, J. (Eds.): Advances in Neural Information Processing Systems 34 (NeurIPS 2021). Curran Associates, Inc., 2021, pp. 24261–24272, doi: 10.48550/arXiv.2105.01601.

[3] Touvron, H.—Bojanowski, P.—Caron, M.—Cord, M.—El-Nouby, A.—Grave, E.—Izacard, G.—Joulin, A.—Synnaeve, G.—Verbeek, J.—Jégou, H.: ResMLP: Feedforward Networks for Image Classification with Data-Efficient Training. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2022, No. 4, pp. 5314–5321, doi: 10.1109/TPAMI.2022.3206148.

[4] Liu, S.—Wang, L.—Yue, W.: An Efficient Medical Image Classification Network Based on Multi-Branch CNN, Token Grouping Transformer and Mixer MLP. Applied Soft Computing, Vol. 153, 2024, Art. No. 111323, doi: 10.1016/j.asoc.2024.111323.

[5] Chen, S.—Xie, E.—Ge, C.—Chen, R.—Liang, D.—Luo, P.: CycleMLP: A MLP-Like Architecture for Dense Prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2023, No. 12, pp. 14284–14300, doi: 10.1109/TPAMI.2023.3303397.

[6] Tang, C.—Zhao, Y.—Wang, G.—Luo, C.—Xie, W.—Zeng, W.: Sparse MLP for Image Recognition: Is Self-Attention Really Necessary? Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, No. 2, pp. 2344–2351, doi: 10.1609/aaai.v36i2.20133.

[7] Tu, Z.—Talebi, H.—Zhang, H.—Yang, F.—Milanfar, P.—Bovik, A.—Li, Y.: MAXIM: Multi-Axis MLP for Image Processing. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 5759–5770, doi: 10.1109/CVPR52688.2022.00568.

[8] Liu, H.—Dai, Z.—So, D. R.—Le, Q. V.: Pay Attention to MLPs. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., Wortman Vaughan, J. (Eds.): Advances in Neural Information Processing Systems 34 (NeurIPS 2021). Curran Associates, Inc., 2021, pp. 9204–9215, doi: 10.48550/arXiv.2105.08050.

[9] Fusco, F.—Pascual, D.—Staar, P.—Antognini, D.: pNLP-Mixer: An Efficient All-MLP Architecture for Language. In: Sitaram, S., Klebanov, B. B., Williams, J. D. (Eds.): Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track) (ACL 2023). 2023, pp. 53–60, doi: 10.18653/v1/2023.acl-industry.6.

[10] Mai, F.—Pannatier, A.—Fehr, F.—Chen, H.—Marelli, F.—Fleuret, F.—Henderson, J.: HyperMixer: An MLP-Based Low Cost Alternative to Transformers. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (Eds.): Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2023). 2023, pp. 15632–15654, doi: 10.18653/v1/2023.acl-long.871.

[11] Tang, H.—Wang, Y.—Li, R.: LCP-Mixer: A Lightweight Model Based on Concept-Level Perception for NLP. International Journal of Data Science and Analytics, Vol. 20, 2025, No. 3, pp. 2163–2173, doi: 10.1007/s41060-024-00588-9.

[12] Liu, X.—Tang, H.—Zhao, J.—Dou, Q.—Lu, M.: TCAMixer: A Lightweight Mixer Based on a Novel Triple Concepts Attention Mechanism for NLP. Engineering Applications of Artificial Intelligence, Vol. 123, Part C, 2023, Art. No. 106471, doi: 10.1016/j.engappai.2023.106471.

[13] Tang, H.—Liu, X.—Wang, Y.—Dou, Q.—Lu, M.: Pay Attention to the Hidden Semanteme. Information Sciences, Vol. 640, 2023, Art. No. 119076, doi: 10.1016/j.ins.2023.119076.

[14] Tang, H.—Wang, Y.—Zhang, Y.—Dou, Q.—Lu, M.: TS-Mixer: A Lightweight Text Representation Model Based on Context Awareness. Expert Systems, Vol. 42, 2025, No. 2, Art. No. e13732, doi: 10.1111/exsy.13732.

[15] Blei, D. M.—Ng, A. Y.—Jordan, M. I.: Latent Dirichlet Allocation. Journal of Machine Learning Research, Vol. 3, 2003, pp. 993–1022, `https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf`.

[16] Li, H.—Arora, A.—Chen, S.—Gupta, A.—Gupta, S.—Mehdad, Y.: MTOP: A Comprehensive Multilingual Task-Oriented Semantic Parsing Benchmark. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (Eds.): Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL 2021). 2021, pp. 2950–2962, doi: 10.18653/v1/2021.eacl-main.257.

[17] Devlin, J.—Chang, M. W.—Lee, K.—Toutanova, K.: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL 2019), 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[18] Katz, D. M.—Bommarito, M. J.—Gao, S.—Arredondo, P.: GPT-4 Passes the Bar Exam. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 382, 2024, No. 2270, Art. No. 20230254, doi: 10.1098/rsta.2023.0254.

[19] Jiao, X.—Yin, Y.—Shang, L.—Jiang, X.—Chen, X.—Li, L.—Wang, F.—Liu, Q.: TinyBERT: Distilling BERT for Natural Language Understanding. In: Cohn, T., He, Y., Liu, Y. (Eds.): Findings of the Association for Computational Linguistics (EMNLP 2020). 2020, pp. 4163–4174, doi: 10.18653/v1/2020.findings-emnlp.372.

[20] Sun, Z.—Yu, H.—Song, X.—Liu, R.—Yang, Y.—Zhou, D.: MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.): Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). 2020, pp. 2158–2170, doi: 10.18653/v1/2020.acl-main.195.

[21] Kaliamoorthi, P.—Ravi, S.—Kozareva, Z.: PRADO: Projection Attention Networks for Document Classification On-Device. In: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.): Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). 2019, pp. 5012–5021, doi: 10.18653/v1/D19-1506.

[22] Kaliamoorthi, P.—Siddhant, A.—Li, E.—Johnson, M.: Distilling Large Language Models into Tiny and Effective Students Using pQRNN. 2021, doi:

10.48550/arXiv.2101.08890.

[23] Liu, Y.—Ott, M.—Goyal, N.—Du, J.—Joshi, M.—Chen, D.—Levy, O.—Lewis, M.—Zettlemoyer, L.—Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019, doi: 10.48550/arXiv.1907.11692.

[24] Yang, Z.—Dai, Z.—Yang, Y.—Carbonell, J.—Salakhutdinov, R. R.—Le, Q. V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). 2019, pp. 5753–5763, doi: 10.48550/arXiv.1906.08237.

[25] Sun, C.—Qiu, X.—Xu, Y.—Huang, X.: How to Fine-Tune BERT for Text Classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (Eds.): Chinese Computational Linguistics (CCL 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11856, 2019, pp. 194–206, doi: 10.1007/978-3-030-32381-3_16.

[26] Beltagy, I.—Peters, M. E.—Cohan, A.: Longformer: The Long-Document Transformer. CoRR, 2020, doi: 10.48550/arXiv.2004.05150.

[27] Lee-Thorp, J.—Ainslie, J.—Eckstein, I.—Ontañón, S.: FNet: Mixing Tokens with Fourier Transforms. In: Carpuat, M., de Marneffe, M. C., Meza Ruiz, I. V. (Eds.): Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022). 2022, pp. 4296–4313, doi: 10.18653/v1/2022.naacl-main.319.

Shuobin Zhang graduated with his Master's degree from the School of Computer Science and Technology at the Shandong Technology and Business University. His research interests include machine learning, artificial intelligence and data mining.

Jianhua Sun is currently working at the Weihai Municipal Hospital, Cheeloo College of Medicine, Shandong University. His academic pursuits are centered around the fields of machine learning, artificial intelligence, and data mining.

Huanling Tang received her B.Sc. in the Yantai University in 1993, her M.Sc. degree from the Tsinghua University in 2004, and her Ph.D. degree from the Dalian Maritime University in 2009. Now she is Professor in the School of Computer Science and Technology at the Shandong Technology and Business University. Her research interests include machine learning, artificial intelligence and data mining.

Wenhao Duan graduated with his Master's degree from the School of Information and Electronic Engineering at the Shandong Technology and Business University. His research interests include federated learning and artificial intelligence.

Quansheng Dou received his M.Sc. and Ph.D. degrees from the Jilin University in 2001 and 2005. Now he is Professor in the School of Computer Science and Technology at Shandong Technology and Business University. His research interests include machine learning, artificial intelligence and evolutionary computation.

Mingyu Lu received his M.Sc. and Ph.D. degrees in computer science and technology from the Tsinghua University in 1988 and 2002, respectively. Now he is Professor at the Dalian Maritime University. His research interests include machine learning, artificial intelligence and data mining.