Improving Personalized Sentiment Representation with Knowledge-enhanced and Parameter-efficient Layer Normalization

You Zhang[1], **Jin Wang**[1,*], **Liang-Chih Yu**[2,*], **Dan Xu**[1], **Xuejie Zhang**[1]
[1]School of Information Science and Engineering, Yunnan University, Yunnan, P.R.China
[2]Department of Information Management, Yuan Ze University, Taiwan
{yzhang0202, wangjin}@ynu.edu.cn, lcyu@saturn.yzu.edu.tw

Abstract

Existing studies on personalized sentiment classification consider a document review as an overall text unit and incorporate backgrounds (i.e., user and product information) to learn sentiment representation. However, it is difficult when these methods meet the current pretrained language models (PLMs) owing to quadratic costs that increase with text length and heterogeneous mixes of randomly initialized background information and textual information initialized from well-pretrained checkpoints during information incorporation. To address these problems, we propose a knowledge-enhanced and parameter-efficient layer normalization (E2LN) for efficient and effective review modeling via leveraging LN in transformer structure. Initially, a knowledge base is introduced that stores *well-pretrained checkpoints*, *structured text information*, and *background information*. Based on such a knowledge base, the ability of LN can be magnified as being a crucial component of transformer structure and then improve the performance of PLMs in downstream tasks. Moreover, the proposed E2LN can make PLMs capable of modeling long document reviews and incorporating background information with parameter-efficient fine-tuning and knowledge injecting. Extensive experimental results were obtained for three document-level sentiment classification benchmark datasets. By comparing the results, the effectiveness and efficiency of the proposed model was demonstrated. Code and Data are released at https://github.com/yoyo-yun/E2LN.

Keywords: Personalized Sentiment Analysis, Layer Normalization, Pretrained Language Model

1. Introduction

Text sentiment classification and regression aim to automatically determine users' overall sentiment polarities or intensities toward a particular topic or event from texts used to survey user attitudes (Liu, 2012; Poria et al., 2023; Lu et al., 2023; Buechel and Hahn, 2017; Lee et al., 2022). Recently, its commercial potential has significantly increased due to the exponential growth in online reviews on various websites, such as IMDb and Amazon (Fang and Zhan, 2015). Compared with traditional text classification, personalized review sentiment classification requires an intelligent system to identify fine-grained polarities in document-level reviews (e.g., IMDb reviews ratings range 1–10 stars) instead of binary or trinary ones, which can facilitate numerous real-world applications.

One common solution to perform personalization is to introduce external background knowledge, such as user and product (UP) information, usually, in the form of non-textual tokens (Tang et al., 2015; Dong et al., 2017; Amplayo, 2019; Chen et al., 2016; Wu et al., 2018). Most of these models are based on traditional neural networks such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and learn UP information with model optimizations from scratch. Recently, pretrained language models (PLMs) based on transformer structures (Vaswani et al., 2017) have achieved considerable success in sentiment analysis (Zhou et al., 2020; Yuan et al., 2023), such as BERT (Devlin et al., 2019). They outperform traditional neural networks owing to transfer learning, which initializes sentiment models from a well-pretrained checkpoint. Despite continued efforts to improve contextual representation for various tasks in natural language understanding, using transformers to learn personalized sentiment representations is still difficult. The key challenges are mainly twofold:

(1) The computational complexity in transformers increases at a quadratic rate with the input text length (Beltagy et al., 2020). As a result, a series of PLMs limit the maximum length to avoid overwhelmed deployments (Devlin et al., 2019; Liu et al., 2019). To address this issue, several methods have been adopted for efficiently modeling long documents (Wu et al., 2020; Tay et al., 2020; Wang et al., 2020). These methods are primarily divided into two categories: hierarchical approach (Zhang et al., 2019; Yang et al., 2020) and sparse attention matrix approach (Beltagy et al., 2020; Zaheer et al., 2020). However, neither can fully model the global context of documents and may have suboptimal performance in document-level review modeling tasks (Wu et al., 2021).

(2) Pretrained checkpoints in transformer-based models are agnostic to background information because only textual languages are encoded in the pretraining phase. Accordingly, the heterogeneous mixes of textual information from well-pretrained

*Corresponding authors

LREC-COLING 2024, pages 8877–8889

checkpoints and randomly initialized non-textual information make background information hard to inject directly into PLMs. Although introducing different knowledge injection modules can facilitate the fusion of textual and non-textual information in the fine-tuning phase, it usually relies on sophisticated structure designs and large external parameters to adapt the original structures via fully model fine-tuning (FFT). For example, Zhang et al. (2021a) proposed a multi-attribute attention (MAA) module where 6 additional UP-specific transformer layers stack over language-specific PLM checkpoints, relatively making a large computational budget.

To address the aforementioned problems, this study proposes a knowledge-enhanced and parameter-efficient layer normalization (E2LN) method over transformers, which mainly contains two modules, including LN-based attentive pooling (LNAP) and personalized LN (PLN), for efficiently and effectively review modeling.

Regarding knowledge enhancements, the proposed method is based on the knowledge base which preliminarily contains three parts: 1) Off-the-shelf well-pretrained checkpoints. They initialize the parameters of the transformer structures for downstream tasks; 2) Structured text information. They are generated from chunk-wise hierarchical texts via the LNAP and perform global textual knowledge to facilitate models modeling long document review texts; 3) Background information (i.e., UP). They are zero-initialized as background embedding and are then injected into transformer layers via the PLN for personalization rendering. Here, structure text information can be considered as homogeneous information since they derive from text information by attentive integration (i.e., LNAP). Zero-initialized information in PLN cannot influence textual information from checkpoints at the first step in the fine-tuning stage thus it gradually facilitates heterogenous information incorporation during further fine-tuning (Zhang et al., 2023).

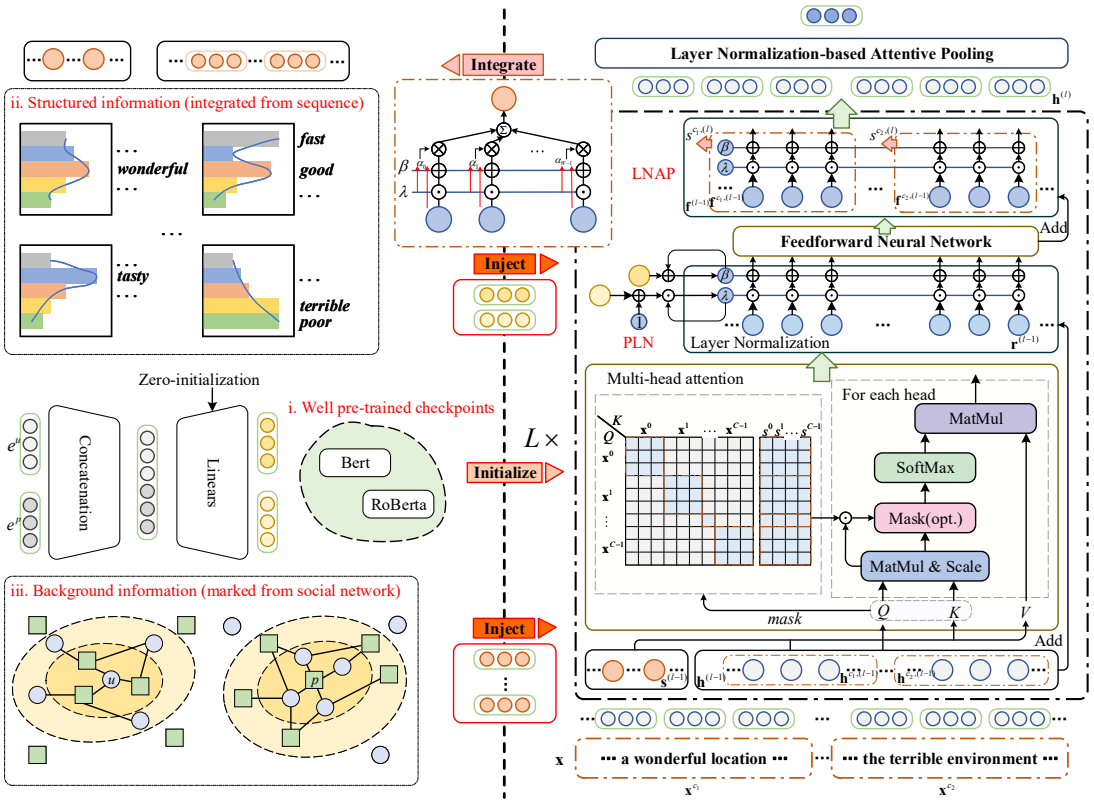
Moreover, the proposed E2LN is parameter-efficient since: 1) We propose a straightforward but efficient LN-tuning (LNT) of parameter-efficient fine-tuning (PEFT) with gain and bias terms of LN to bridge pretraining and fine-tuning stages; 2) LNAP is external-parameter-free where only original gain and bias terms of LN are used for attentive calculation; 3) In comparison with previous knowledge injection methods such as multi-attribute attention (MAA) (Zhang et al., 2021a), PLN is much lighter due to vector-shaped gain and bias parameters accommodating UP information. The features of parameter efficiency in E2LN can also shed light on large PLMs, such as Flan-T5 (Chung et al., 2022) and LLaMA (Touvron et al., 2023), performing personalized sentiment analysis and other personalized services due to the same transformer structures all of they used.

Extensive experiments were conducted on three benchmark datasets: IMDB, Yelp-2013, and Yelp-2014 (Tang et al., 2015). Experimental results demonstrated that the proposed model yield on-par or better performance compared to previous high-performance models even with fewer trainable parameters. Additionally, an ablation study and complexity analysis reveal the effect and high efficiency of the proposed method in personalized sentiment analysis.

2. Related Work

Personalized Sentiment Analysis. Text sentiment analysis is intended to automatically determine the attitudes of people toward a certain target natural language text. To identify exact sentiments, personalized sentiment analysis usually uses personalized background information, such as UP information, over long-document reviews (Tang et al., 2015; Amplayo, 2019; Zhang et al., 2021a). Background information is generally collected from social networks, such as Amazon and IMDb, identified with token IDs. However, most existing studies consider each of them as an overall text unit and then use traditional neural networks (e.g., LSTM) to perform sentiment analysis (Bermingham and Smeaton, 2011; Kim, 2014; Chung et al., 2014). To address this problem, a hierarchical attentive network (HAN) has been proposed for personalized sentiment analysis (Yang et al., 2016; Chen et al., 2016; Wu et al., 2018). To facilitate transformer modeling for long documents, two methods have been proposed: hierarchical structure (e.g., Hierarchcial BERT (Zhang et al., 2019)) and sparse attention (e.g., Lonformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020)). However, these models cannot fully model global document contexts (Wu et al., 2021).

Normalization. It is one of the most significant components in neural networks that can normalize representations to obtain smooth gradients, fast learning, and improved generalization. For example, batch normalization (Ioffe and Szegedy, 2015), layer normalization (Ba et al., 2016), and group normalization (Wu et al., 2020) are in normalization family. Furthermore, recent research has discovered that normalization can be extended to broader applications, such as style transfer (Lee et al., 2021; Sun et al., 2021) and recognizing salience information (Liu et al., 2017; yichao liu et al., 2021). These models mentioned that the gain and bias terms in normalizations were trainable parameters for scaling and shifting and the gain (or scaling) term can further present importance of features. This study extended LN to accomondate background information and proposed an LNAP to capture structured



(a) Knowledge Base (b) E2LN-based Transformer Structure

Figure 1: Framework of E2Transformer.

text information. Compared with previous studies, the proposed method does not require additional modules, such as fully connected networks and CNNs, to construct injection and attention modules.

Parameter-efficient Fine-tuning. PEFT provides a solution to alleviate the cost of FFT for PLMs with ever-growing sizes and demonstrate on-par performance as FFT. Currently, PEFT methods are mainly divided into two categories: 1) sparse tuning methods, that fine-tune a small of parameters in well-pretrained checkpoints (Guo et al., 2021; Sung et al., 2021; Tay et al., 2020; Zhou et al., 2020). For example, Ben Zaken et al. (2022) proposed a BitFit method which only tunes bias-term parameters in PLMs for downstream task adaptation; 2) adding and fine-tuning a relatively small number of parameters, including adapter (Pfeiffer et al., 2020), a low-rank version of adapter dubbed LoRA (Hu et al., 2021), P-tuning (Liu et al., 2023), prompt-tuning (Lester et al., 2021), and prefix tuning (Li and Liang, 2021). Closest to our method, Qi et al. (2022) indicates LNT is a viable PEFT method effectively and efficiently against the gap between pretraining and fine-tuning phases and orthogonal to other PEFT methods. The main difference is that our proposed method is not only for downstream tasks adaptation but also for knowledge extraction and injection.

3. Methodology

Figure 1 shows the framework of E2Transformer that aims to learn a robust personalized review representation. It primarily comprises two parts: a knowledge base and an E2Transformer structure or E2LN-based transformer structure. The knowledge base activates the structure for efficiently and effectively handling long document reviews.

An online review is usually a long textual document $\mathbf{x}$ with a fine-grained rating y and background information $\mathbf{b}$. Personalized sentiment classification task requires a sentiment model $f(\mathbf{x}, \mathbf{b})$ that takes the text $\mathbf{x}$ and background information $\mathbf{b}$ as inputs and automatically learns to determine the sentiment $\hat{y}$, which is expected to be close to the golden rating y. Here, background information is assigned special IDs to present specific domains. For example, we use a pair of user ID u and product ID p to indicate a review written by the user u toward the product p, i.e., $\mathbf{b} = \{u, p\}$.

3.1. The Knowledge Base

Well-pretrained Checkpoints. They are off-the-shelf weights in transformers learned from a large number of general texts and are then fine-tuned for downstream tasks as general knowledge.

Background Information. To represent UP information, a special word-embedding technique is adopted to convert discrete UP IDs into dense ones, denoted as $e^u \in \mathbb{R}^{d_u}$ and $e^p \in \mathbb{R}^{d_p}$, respectively. Background representation can render text representation in transformer structures in forward propagation and update itself during backward learning.

Structured Text Information. It is constructed from textual sequence representation via LNAP using a sliding window, which stores global document contexts and can be easily injected into transformer structures for document modeling.

3.2. E2Transformer Structure

Similar to the transformer structure (Vaswani et al., 2017), we adopted a well pretrained tokenizer to split a given text into discrete N tokens $\mathbf{x} = \{x_0, x_1, \cdots, x_{N-1}\}$. Before $\mathbf{x}$ is fed into the transformer layer, it is word-embedded and added with positional embeddings, denoted as $\mathbf{h}^0 = \{h_0^0, h_1^0, \cdots, h_{N-1}^0\} \in \mathbb{R}^{N \times d_h}$, with d_h dimensionality as well as the input of the first encoder layer. To efficiently model long documents, we also view the sequence using a sliding window of size W to segment the entire text into $C = N/W$ chunks $\mathbf{h}^{(0)} = \{\mathbf{h}^{0,(0)}, \mathbf{h}^{1,(0)}, \cdots, \mathbf{h}^{C-1,(0)}\} \in \mathbb{R}^{C \times W \times d_h}$, where the zero-padding method is used to ensure exact division.

Every E2Transformer layer (E2Layer) has a structure similar to the vanilla transformer encoder layer, mainly containing four components: multi-head attention (MHA), PLN, feedforward neural network (FFN), and LNAP. Compared with the original transformer encoder layer, E2Layer is only technically knowledge-enhanced and parameter-efficient. Subsequently, we introduce these components following information forward propagations.

Multi-head Attention. MHA takes as layer input $\mathbf{h}^{(l-1)}$ and updates the token representation through interactions within sequential tokens, where $l \in [1 : L]$ refers to the lth layer in an E2Transformer structure with L layers.

The MHA first maps the inputs into queries, keys, and values via linear projections, denoted as $Q = [Q^c]_{c=0}^{C-1} \in \mathbb{R}^{C \times W \times d_h}$, $K = [K^c]_{c=0}^{C-1} \in \mathbb{R}^{C \times W \times d_h}$, and $V \in \mathbb{R}^{N \times d_h}$, respectively. It then computes the relatedness $\mathbf{a}$ between Q and K in each chunk using scaled-dot product alignment functions. For each chunk c, $\mathbf{a}$ is formulated as

$$\mathbf{a}^c = (Q^c \cdot {K^c}^\top)/\sqrt{d_h} \in \mathbb{R}^{W \times W} \tag{1}$$

Next, chunk-wise attention scores are mapped onto full-range attention maps in a sparse format filled with zero paddings. The final attention score is denoted as

$$\mathbf{a} = \mathrm{softmax}(\begin{bmatrix} \mathbf{a}^0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{a}^1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{a}^{C-1} \end{bmatrix} \odot mask(\mathbf{x})) \tag{2}$$

where $mask(\cdot)$ is the function for generating input masks, which then calculates Hadamard products ($\odot$) with sparse attention scores to ignore useless padding tokens in attention interactions. Finally, V is multiplied by sparse attention to update the sequential representation:

$$\tilde{\mathbf{h}} = \mathbf{a} \cdot V \in \mathbb{R}^{N \times d_h} \tag{3}$$

Based on multi-head mechanism, all headwise vectors are concatenated into a dense vector and then projected into a comprehensive vector $\mathbf{r} \in \mathbb{R}^{N \times d_h}$ with d_h dimensionality.

To model the aforementioned local tokens in each chunk using global document contexts, *structured text information* $\mathbf{s}^{(l-1)} \in \mathbb{R}^{C \times d_h}$ in the knowledge base is extended to E2Transformer input $\mathbf{h}^{(l-1)}$, generating structure-enhanced keys by concatenating $\mathbf{s}$ at the end of each chunk $\mathbf{h}^c$ ($\{[\mathbf{h}^c; \mathbf{s}]\}_{c=0}^{C-1}$) and structure-enhanced values by concatenating $\mathbf{s}$ at the end of the sequence ($[\mathbf{h}; \mathbf{s}]$), denoted as $K^{Se} \in \mathbb{R}^{C \times (W+C) \times d_h}$ and $V^{Se} \in \mathbb{R}^{(N+C) \times d_h}$, respectively, where $[;]$ denotes the concatenation method. To this end, the structure-enhanced attention score for each chunk $\mathbf{a}^{Se,c} \in \mathbb{R}^{W \times (W+C)}$ is calculated using Eq. (1) with Q^c and $K^{Se,c}$ as inputs. Furthermore, $\mathbf{a}^{Se}$ and $mask([\mathbf{x}; \mathbf{s}])$ are mapped in the shape of $N \times (N+C)$ using Eq. (2), as shown in Figure 1. Consequently, structure-enhanced outputs $\tilde{\mathbf{h}}^{Se} \in \mathbb{R}^{N \times d_h}$ are generated using Eq. (3) with $\mathbf{a}^{Se}$ and V^{Se} as inputs, similar to $\tilde{\mathbf{h}}$ in the dimensionalities. Therefore, the structure-enhanced representation of the MHA output $\mathbf{r}^{Se} \in \mathbb{R}^{N \times d_h}$ is successively captured in multi-head mechanism.

Personalized Layer Normalization. Based on LN in the transformer, we propose a PLN to render the token representation $\mathbf{r}^{Se} = [r_0^{Se}, r_1^{Se}, \cdots, r_{N-1}^{Se}]$, generated from MHA, with *personalized background information* of e^u and e^p.

Before the introduction of PLN, LN is first described. Let r_n and ${r_n}' \in \mathbb{R}^{h_d}, n \in [0 : N-1]$ denote token representations before and after LN operations, respectively.

$$\begin{aligned} {r_n}' &= \mathrm{LN}(r_n; \lambda, \beta) = \frac{r_n - \mu_n}{\sigma_n} \odot \lambda + \beta \\ \mu_n &= \frac{1}{d_h} \sum_{i=1}^{d_h} r_{ni} \\ \sigma_n &= \sqrt{\frac{1}{d_h} \sum_{i=1}^{d_h} (r_{ni} - \mu_n)^2} \end{aligned} \tag{4}$$

where μ_n and σ_n denote the mean and standard deviations of input r_n, respectively, r_{ni} denotes the ith dimension of r_n, and $\lambda \in \mathbb{R}^{d_h}$ and $\beta \in \mathbb{R}^{d_h}$ denote the affine transformation (i.e., gain and bias, respectively) parameters for power preservation, obtained by rescaling and recentering the normalized representation (Wu et al., 2020).

The PLN injects background information by personalizing the gain and bias parameters in the original transformer by further rescaling and recentering them, respectively, as

$$\begin{aligned} \lambda^{Pe} &= (\mathbf{1} + \text{linear}_{\lambda}([e^u; e^p])) \odot \lambda \\ \beta^{Pe} &= \text{linear}_{\beta}([e^u; e^p]) + \beta \end{aligned} \tag{5}$$

where $linear(\cdot)$ denotes a one-layer linear projection fusing user and product information. In practice, much more complex structures (e.g., multilayer perceptron or MLP, deep convolutional neural networks or CNN, and graph neural networks) or simpler method (e.g., concatenation, addition, and weighted addition) can also be deployed. Using these operations, a personalization-enhanced representation $\mathbf{r}^{Pe} = [r_n^{Pe}]_{n=0}^{N-1}$ is generated as follows:

$$\begin{aligned} r_n^{Pe} &= \text{PLN}(r_n^{Se} + h_n, e^u, e^p) \\ &= \text{LN}(r_n^{Se} + h_n; \lambda^{Pe}, \beta^{Pe}) \end{aligned} \tag{6}$$

Note that $[e^u; e^p]$ is zero-initialized to protect pretrained checkpoints from the noise that occurs owing to the random initialization of background information in the fine-tuning phase. Thus, it can facilitate the fusion of textual and personalized information where they are located in heterogenous distributions.

Feedforward Neural Network. The FFN was constructed using two-layer linear projections with a rectified linear unit (ReLU) activation function. Through an FFN, the sequential representation is further encoded as follows:

$$\mathbf{f} = \text{FFN}(\mathbf{r}^{Pe}) + \mathbf{r}^{Pe} \in \mathbb{R}^{N \times d_h} \tag{7}$$

where a residual network is also adopted to connect the representations.

Layer Normalization-based Attentive Pooling. LNAP uses the sliding window mechanism to view the representation **f** in the shape of $C \times W \times d_h$ and generates structured text information stored in the knowledge base as global document context, maintaining the original power of the normalization mechanism.

Previous studies revealed that the scale factor (gain) parameters in LN can reflect salience information in a representation vector (Guo et al., 2021; Sung et al., 2021). Inspired by this, LNAP is proposed to structure text information in each chunk, which is formulated as follows:

$$\begin{aligned} score_w^c &= score(f_w^c) = \frac{f_w^c - \mu_w^c}{\sigma_w^c} \lambda^\top \\ \alpha_w^c &= \frac{\exp(score_w^c)}{\sum_{w=0}^{W-1} \exp(score_w^c)} \odot mask(\mathbf{x}^c) \\ s^c &= \sum_w \alpha_w^c \cdot \text{LN}(f_w^c; \lambda, \beta) \in \mathbb{R}^{d_h} \end{aligned} \tag{8}$$

Through concatenation, structured text information over chunks is integrated into the knowledge base, which is denoted as $\mathbf{s}' = [\mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^{C-1}] \in \mathbb{R}^{C \times d_h}$. Furthermore, $\mathbf{f}$ is input into Eq. (4) to calculate the sequence representation $\mathbf{h}'$. Both $\mathbf{s}'$ and $\mathbf{h}'$ (i.e., $\mathbf{h}^{(l)}$ and $\mathbf{s}^{(l)}$) are the inputs for the following $(l+1)$th transformer layer.

Training Objective. After the knowledge enhancements, the E2transformer efficiently models long document reviews and effectively captures robust sequence representations via L layer propagations. To generate the document representation, LNAP is applied in the final layer. Subsequently, we use a linear projection with a softmax activation function as a classifier to predict sentiment distributions.

Inspired by the current PEFT methods, we further adopt LNT method for computational costs saving. In detail, LNT keeps only the gain and bias terms of LN trainable. However, in the experiments, we found that pure LNT (or LNT only) could effectively perform UP injection but hard to promote adaptation in our tasks. To overcome these limitations, we combine LNT with current PEFT methods (i.e., LoRA, BitFit and other sparse fine-tuning) for improvements, seeing Section 4.2 and Appendices B.2.

4. Experiments

4.1. Experiment Settings

Dataset and Metrics. We introduced three traditional document-level sentiment datasets (including IMDB, Yelp-2013, and Yelp-2014) and two convincing metrics (accuracy (Acc %) and root mean squared error (RMSE)), following prior works on personalized backgrounds (Wu et al., 2018). Further statistics of the datasets and detailed implementation descriptions were presented in Appendices A.

Baselines. We introduced current baselines to compare with our method E2LN, with three groups:

(1) **Backbones**. Conventional neural networks included CNN (Kim, 2014) and bidirectional LSTM (BiLSTM) (Sachan et al., 2019); prevalent PLMs from BERT family included BERT ($\mathcal{B}$) (Devlin et al., 2019) and RoBERT ($\mathcal{R}$) (Liu et al., 2019).

(2) **Long dependency information**. These models meant utilizing hierarchical structure or

Models		IMDB		Yelp-2013		Yelp-2014	
		Acc	RMSE	Acc	RMSE	Acc	RMSE
Backbones	CNN	40.5	1.629	57.7	0.812	58.5	0.808
	BiLSTM	43.3	1.494	58.4	0.764	59.2	0.733
	BERT	47.4	1.379	66.0	0.699	66.9	0.622
	RoBERTa	49.3	1.248	68.9	0.604	69.0	0.606
+ Long Dependency	NSC	44.3	1.465	62.7	0.701	63.7	0.686
	NSC+LA	48.7	1.381	63.1	0.706	63.0	0.715
	ToBERT	50.8	1.194	66.7	0.662	66.9	0.620
	HiBERT	51.7	1.192	67.1	0.632	67.4	0.627
	Longformer ($\mathcal{R}$)	53.6	1.129	69.6	0.586	69.6	0.590
	BigBird ($\mathcal{R}$)	53.7	1.121	69.8	0.585	69.5	0.599
	CK-BERT	52.3	1.194	68.1	0.618	68.1	0.613
	CK-RoBERTa	53.5	1.148	69.0	0.612	69.3	0.603
+ UP Background	UPA (NSC)	53.3	1.281	65.0	0.692	66.7	0.654
	UAPA (NSC)	55.0	1.185	68.3	0.628	68.6	0.626
	IAA (NSC)	56.4	1.158	-	-	69.4	0.621
	CHIM (BiLSTM)	56.4	1.161	67.8	0.646	69.2	0.629
	MAA (CK-$\mathcal{B}$)	57.3	1.042	70.3	0.588	71.4	0.573
	MAA (CK-$\mathcal{B}$)†	57.2	1.050	70.0	0.593	71.4	0.587
	MAA (CK-$\mathcal{R}$)†	58.3	1.015	71.6	0.562	72.5	0.567
	MAA ($\mathcal{B}$)†	53.0	1.141	69.3	0.594	70.0	0.579
	MAA ($\mathcal{R}$)†	54.8	1.074	71.5	0.578	72.4	0.565
Ours (FFT)	E2LN ($\mathcal{B}$)	58.4	1.050	70.4	0.586	71.4	0.571
	E2LN ($\mathcal{R}$)	**59.8**	**0.972**	71.9	**0.562**	**73.0**	**0.555**
+ PEFT	E2LN ($\mathcal{B}$) LNT	44.8	1.158	64.6	0.676	65.1	0.674
	E2LN ($\mathcal{R}$) LNT	48.9	1.119	68.0	0.625	68.4	0.605
	E2LN ($\mathcal{B}$) MHA + LNT	58.4	1.052	70.3	0.595	71.3	0.582
	E2LN ($\mathcal{R}$) MHA + LNT	**59.8**	0.959	**72.1**	**0.562**	**73.0**	0.556

Table 1: Results of the proposed and baseline models. The **boldface** figures denoted the best results among all methods and underscored figures denoted the best baseline results among each group. All results were averaged over five runs. † especially denoted performance reimplementation from authors' original codes under the same experimental environments as ours.

Models	IMDB	Yelp-2013	Yelp-2014
ours E2LN ($\mathcal{B}$)	58.4	70.4	71.4
w/o U	52.0	67.3	67.7
w/o P	57.1	69.6	69.3
w/o UP	51.9	67.1	67.4
w/o Se	52.7	69.3	70.3

Table 2: Ablation study of accuracy on E2LN ($\mathcal{B}$).

sparse attention for long document modeling, including: NSC with local attention (LA) (Chen et al., 2016), transformer over BERT (ToBERT) (Pappagari et al., 2019), hierarchical BERT (HiBERT) (Zhang et al., 2019), Longformer (Beltagy et al., 2020) and BidBird (Zaheer et al., 2020). Moreover, a cherry pick (CK) truncate strategy[1] used in (Zhang et al., 2021a) and (Sun et al., 2019) was applied to pick up length-limited tokens for PLMs that restricted the maximum input length to avoid quadratic costs increasing.

(3) **UP background information**. To perform personalization, a series of UP injection methods were introduced, including: user and product attention (UPA), user attention and product attention (UAPA), interactive attribute attention (IAA) (Zhang et al., 2021b), MAA (Zhang et al., 2021a), and CHIM (Amplayo, 2019).

4.2. Experiment Results

Comparative Results. The main results for all methods were listed in Table 1. Regarding different backbones, models performed to different extents where PLMs outperformed traditional neural networks.

Compared with the first group, the second group of methods relatively achieved better performance on all three datasets, where, for example, NSC vs. BiLSTM; ToBERT or HiBERT vs. BERT. Notably, it could be found that with different truncate strategies (CK and direct truncate[2]), models performed with different results, especially for IMDB dataset. This phenomenon indicted truncate strategies were suboptimal since input tokens were empirically selected, ignoring full consideration of complete re-

[1]The first 128 tokens concatenate the last 384 tokens, empirically.

[2]Remaining the first 512 tokens in document-level texts.

	Models	IMDB	Yelp-2013	Yelp-2014
	E2LN ($\mathcal{B}$)			
PLN	w/ PLN (after MHA) only	58.4	70.4	71.4
	w/ PLN (after FFN) only	57.5	70.5	71.4
	w/ PLN (after MHA) & PLN (after FFN)	57.6	70.2	71.2
Module	+ MHA	57.8	70.3	70.7
	+ FFN	58.6	70.0	70.8
	+ MHA & FFN	57.8	69.8	70.8
	w/ MHA only	58.1	70.3	70.9
	w/ FFN only	57.8	70.1	70.6
Layer	1-6 layers only	57.4	70.2	70.7
	7-12 layers only	58.2	70.2	71.1

Table 3: Accuracy of E2LN ($\mathcal{B}$) for the investigation of UP injections. **PLN** means UP injections at different LNs. **Module** and **Layer** denote other modules (not matrix but only bias terms) and layers in the transformer structure activated for UP injection, respectively. w/ means only corresponding places where UP is injected, and + presents additional injections utilized based on the proposed E2LN.

view information. Longformer ($\mathcal{R}$) and BigBird ($\mathcal{R}$) could achieve better results in the second group because they introduced sparse attentions to tackle long documents. Unfortunately, they failed to directly enable PLMs such as BERT and RoBERTa to model long documents over 512 tokens, requiring further pretraining via warm-starting from RoBERTa checkpoints on a large amount of corpus.

Incorporating with background information of UP, the third group of methods achieved much higher scores of Acc and lower figures of RMSE. These findings revealed the importance of UP in personalized sentiment analysis. From traditional backbones to PLMs, most of current personalized sentiment models have redesigned sophisticated structures. Regarding MAA (CK-$\mathcal{B}$), it removed information-unpredictable tokens for satisfying the input limitations of PLMs. By contrast, the proposed E2LN achieved the best performance in a more parameter-efficient UP injection way, demonstrating its positive effect in personalized sentiment classification tasks.

Moreover, it can be observed that pure LNT (or LNT only) degraded the performance on all three datasets. A possible reason might be that although LN had shown its effectiveness in UP injections, fine-tuning LN alone was not enough for task adaptation in sentiment analysis. However, combing with previous PEFT methods such as sparse fine-tuning of MHA (i.e., MHA+LNT) gained on-par personalized sentiment analysis performance than FFT. More dynamic combinations between LNT and other PEFT methods for personalized sentiment were reported in Appendices B.2.

Ablation Study. To validate the effectiveness of the proposed E2LN method, an ablation experiment were conducted in Table 2. Firstly, both Acc and RMSE performance degraded with the elimination of user or product information, demonstrating the effect of personalized background information injection. Furthermore, the elimination of user information affected the performance more than those of product information, indicating that the personalized background information of users was more crucial for sentiment analysis than product information. This was because user information is directly related to the subjective sentiments of users in reviews, whereas product information only contained objective characteristics for certain products. Next, we stopped integrating and injecting structured text information during the training and prediction phases, and the results were consequently lowered. Without structured text information, the BERT-based model can only model the local information of text input within the first 512 tokens, which is not sufficient to effectively handle long-document inputs.

4.3. Analysis of Knowledge Enhancements

UP Injection. In transformer structures, we denoted two LNs deployed after MHA and FFN at each layer as LN (after MHA) and LN (after FFN), respectively. To explore how LN-based injections influence the model performance, the first group in Table 3 reported Acc scores of E2LN ($\mathcal{B}$) with personalized background information injections at different LNs (called PLNs). E2LN ($\mathcal{B}$) obtained the best results when personalized background information was injected into LN (after MHA), but not into LN (after FFN). This may be because LN (after FFN) had been extended to LNAP for structured text information generating (see Figure 1). Both structured and personalized knowledge injections would impose a considerable burden on LN (after FFN) with conflict gradients from multiple optimal objectives.

To further compare and improve the performance of the proposed method with other components injecting UPs, we conducted experiments as shown in 2nd-3rd groups. Specially, we had the following

Models	IMDB	Yelp-2013	Yelp-2014
E2LN ($\mathcal{B}$)			
+AvgP	57.8	69.7	71.1
+MaxP	53.1	66.8	68.6
+AttP	57.9	70.1	71.3
+LNAP	58.4	70.4	71.4

Table 4: Acc Performance (%) of various pooling methods embedded into E2LN ($\mathcal{B}$).

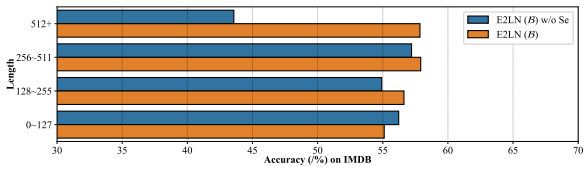


Figure 2: Bar plot for IMDB Dev Acc (Y-axis) on different lengths (X-axis) of E2LN w/o Se and E2LN over BERT.

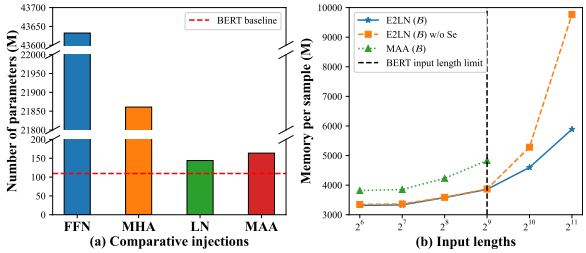


Figure 3: Efficiency analysis for PLN and LNAP on IMDB datasets.

findings. 1) Varying injection modules, different performance gained. Comparing with MHA and FFN based injections, LN based injections achieved relatively better performance. Moreover, it can be found that performance of dynamic combinations of various injections were sensitive to applied datasets and suggested a flexible combination strategy for real-world applications. 2) High-layer (7-12 layers) injections outperformed low-layer (1-6) injections. A possible reason might be that high layers in transformer structures could encode more semantic representation than low layers, which encoded more syntactic information, and semantic UP alignments were more beneficial for personalization.

Effect of LNAP. We separately conducted experiments on E2LN ($\mathcal{B}$) with several other pooling methods, as shown in Table 4. Compared with fixed pooling methods, that is, average pooling (AvgP) and maximum pooling (MaxP), attentive pooling methods, that is, attentive pooling (AttP) and LNAP, achieved better results. This was because attentive pooling methods can dynamically capture salient information over sequences to obtain a robust representation. AttP and LNAP obtained comparable results, demonstrating the effect of LNAP. Compared with AttP in structure, LNAP does not require additional modules with external parameters for fine-tuning.

To further reveal how structured text information facilitates the handling of long documents, we conducted a fine-grained analysis for different input lengths, as shown in Figure 2. Regarding the different input lengths, the different comparative Acc were pictured. Within 512 input tokens, both E2LN w/o Se and E2LN that are initialized from BERT achieved competitive results. When the number of input tokens increased over 512, Acc of E2LN ($\mathcal{B}$) w/o Se sharply decreased, whereas E2LN ($\mathcal{B}$) preserved itself from such damage. This phenomenon demonstrated the effect of the injection of structured text information that was generated from LNAP.

4.4. Analysis of Efficiency

Parameter Efficiency in PLN. In comparison with MHA and FFN, injecting UP into LN is more parameter-efficient. MHA and FFN contain high-dimensional matrix-shaped parameters ($d_{in} \times d_{out}$) requiring dimensionalities of background information (e.g., only for user) $D = N_u \times d_{in} \times d_{out}$ for injection alignments where N_u presents the number of users. While for LN, it only contains vector-shaped parameters (d_v) requiring the dimensionalities of $D = N_u \times d_v$. Here, in BERT checkpoints, there is $d_{in} = d_{out} = d_v = 768$.

For the IMDB dataset, Figure 3(a) showed the parameters requirements for alone user information injection on different modules based on BERT checkpoints. The order of injection modules by requiring the number of external parameters was FFN>MHA>LN, which was the same as the order of the number of their parameters. What's more, it can be found that with 12 PLN equipped, the proposed method was deployed with less parameters than MAA ($\mathcal{B}$) who stacked 6 personalized layers over BERT checkpoints, further demonstrating the efficiency.

Efficiency for Structured Textual Information. In local chunk-wise context modeling, the computational complexity is $O\left(C \cdot W^2 \cdot d\right)$, where C denotes the total number of chunks with size W, and d denotes the dimensionality of hidden states. By injecting global document contexts into local representations, the complexity is calculated as $O\left(C^2 \cdot W \cdot d\right)$. Therefore, the total complexity of the proposed model is $O\left(C \cdot W^2 \cdot d + C^2 \cdot W \cdot d\right)$, which indicates that it is considerably more efficient than the original transformer whose complexity is $O(N^2 \cdot d)$, where N is $C \cdot W$.

In practices, we also analyzed the statistics in occupied training memory that were shown in Figure 3(b). With the input lengths extend, the memory in the training phase was occupied increasingly.

Since MAA ($\mathcal{B}$) introduced additional 6 Transformer layers, it was generally allocated more memories than our methods. With structural knowledge enhanced, the proposed method could break the input length limitation in most of PLMs with a lower increasing trend in memory occupations, consistent with the abovementioned efficiency discussion.

5. Conclusions

In this paper, we proposed the E2LN to effectively and efficiently model personalized reviews. It adopted a knowledge base that contains three aspects to leverage a robust review representation. Experimental results for three document-level sentiment datasets showed that the proposed method outperformed previous high-performance methods, thereby demonstrating its effectiveness and efficiency.

Future work could adopt exact opinion-based structured information, such as aspect-based sentiments with sentiment holders, targets, expressions, and polarity, for knowledge enhancement.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038, 62266051 and 62162068, the Ministry of Science and Technology (MOST), Taiwan, ROC, under Grant No. MOST110-2628-E-155-002, and the Yunnan Postdoctoral Science Foundation under Grant No. C615300504048. The authors would like to thank the anonymous reviewers for their constructive comments.

Bibliographical References

Reinald Kim Amplayo. 2019. Rethinking Attribute Representation and Injection for Sentiment Classification. In *EMNLP-IJCNLP*, pages 5601–5612.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv eprint arXiv:1607.06450*.

S. Behdenna, F. Barigou, and G. Belalem. 2018. Document Level Sentiment Analysis: A survey. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 4(13):154339.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *ACL-2022*, pages 1–9.

Adam Bermingham and Alan Smeaton. 2011. On using Twitter to monitor political sentiment and predict election results. In *SAAIP*, pages 2–10.

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *EACL*, pages 578–585.

Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. 2020. Tiny transfer learning: Towards memory-efficient on-device learning. *arXiv preprint arXiv:2007.11622*.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural Sentiment Classification with User and Product Attention. In *EMNLP*, pages 1650–1659.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *arXiv eprint arXiv:2210.11416*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS: Workshop on Deep Learning*.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *EACL*, pages 623–632.

Xing Fang and Justin Zhan. 2015. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5.

Nishida Toyoaki Fukuhara Tomohiro, Nakagawa Hiroshi. 2007. People, Understanding sentiment of Analysis, from news articles: Temporal sentiment of social events. In *ICWSM*.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-Efficient Transfer Learning with Diff Pruning. In *ACL-IJCNLP 2021*, pages 4884–4896.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, pages 1746–1751.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The Efficient Transformer. In *ICLR*.

Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L. Zhang. 2021. Enhancing Content Preservation in Text Style Transfer Using Reverse Attention and Conditional Layer Normalization. In *ACL-IJCNLP*, pages 93–102.

Lung-Hao Lee, Jian-Hong Li, and Liang-Chih Yu. 2022. Chinese emobank: Building valence-arousal resources for dimensional sentiment analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–18.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP-2021*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *ACL-IJCNLP-2021*, pages 4582–4597.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning Efficient Convolutional Networks Through Network Slimming. In *ICCV*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Qiang Lu, Xia Sun, Yunfei Long, Zhizezhang Gao, Jun Feng, and Tao Sun. 2023. Sentiment analysis: Comprehensive reviews, recent advances, and open challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Raghavendra Pappagari, Piotr Zelasko, Jesus Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical Transformers for Long Document Classification. In *ASRU*, pages 838–844.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A Framework for Adapting Transformers. In *EMNLP*, pages 46–54.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. 2023. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *IEEE Transactions on Affective Computing*, 14(1):108–132.

Wang Qi, Yu-Ping Ruan, Yuan Zuo, and Taihao Li. 2022. Parameter-Efficient Tuning on Layer Normalization for Pre-trained Language Models. *arXiv preprint arXiv:2211.08682*.

Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. In *AAAI*, pages 6940–6948.

Richard Socher, Alex Perelygin, and Jy Wu. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? In *CCL*, pages 194–206.

Zhongkai Sun, Prathusha K Sarma, Yingyu Liang, and William Sethares. 2021. A New View of Multimodal Language Analysis: Audio and Video Features as Text "Styles". In *EACL*, pages 1956–1965.

Yi-Lin Sung, Varun Nair, and Colin A Raffel. 2021. Training Neural Networks with Fixed Sparse Masks. In *NIPS*, volume 34, pages 24193–24205.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *ACL*, pages 1014–1023.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv eprint arXiv:2009.06732*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and Others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint arXiv:1607.08022*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5999–6009.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling. In *ACL-IJCNLP*, pages 848–853.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2020. Lite Transformer with Long-Short Range Attention. In *ICLR*.

Zhen Wu, Xin Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving review representations with user attention and product attention for sentiment classification. In *AAAI*, pages 5989–5996.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuangjing Huang. 2016. Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification. In *EMNLP*, pages 1660–1669.

Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In *CIMK*, pages 1725–1734.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL-HLT*, pages 1480–1489.

yichao liu, Zongru Shao, yueyang Teng, and Nico Hoffmann. 2021. NAM: Normalization-based attention module. In *NeurIPS: Workshop on ImageNet: Past, Present, and Future*.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2023. Encoding syntactic information into transformers for aspect-based sentiment triplet extraction. *IEEE Transactions on Affective Computing*, pages 1–15.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *NIPS*, pages 17283—-17297.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*, pages 5059–5069.

You Zhang, Jin Wang, Liang-Chih Yu, Dan Xu, and Xuejie Zhang. 2023. Domain Generalization via Switch Knowledge Distillation for Robust Review Representation. In *ACL*, pages 12812–12826.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2021a. MA-BERT: Learning Representation by Incorporating Multi-Attribute Knowledge in Transformers. In *ACL-IJCNLP*, pages 2338–2343.

You Zhang, Jin Wang, and Xuejie Zhang. 2021b. Personalized sentiment classification of customer reviews via an interactive attributes attention model. *Knowledge-Based Systems*, 226:107135.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an Efficient Alternative to Finetuning for Pretrained Language Models. In *EMNLP*, pages 2226–2241.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. Bert loses patience: Fast and robust inference with early exit. In *NIPS*, pages 18330–18341.

A. Detailed Experimental Settings

A.1. Datasets

We introduced three traditional document-level sentiment analysis datasets with personalized backgrounds (i.e., discrete UP information), including IMDB, Yelp-2013, and Yelp-2014. The IMDB dataset contained movie reviews rated in the range of 1–10 stars, and the Yelp datasets contained restaurant reviews with 1–5-star ratings. All the

Datasets	#labels	#reviews	#users	#products	#docs/user	#docs/product	Max. words	Avg. words
IMDB	10	84,919	1,310	1,635	64.82	51.94	2,802	431.6
Yelp-2013	5	78,966	1,631	1,633	48.42	48.36	1,643	212.2
Yelp-2014	5	231,163	4,818	4,194	47.97	55.11	1,643	220.1

Table 5: Statistics of IMDB, Yelp-2013, and Yelp-2014 datasets.

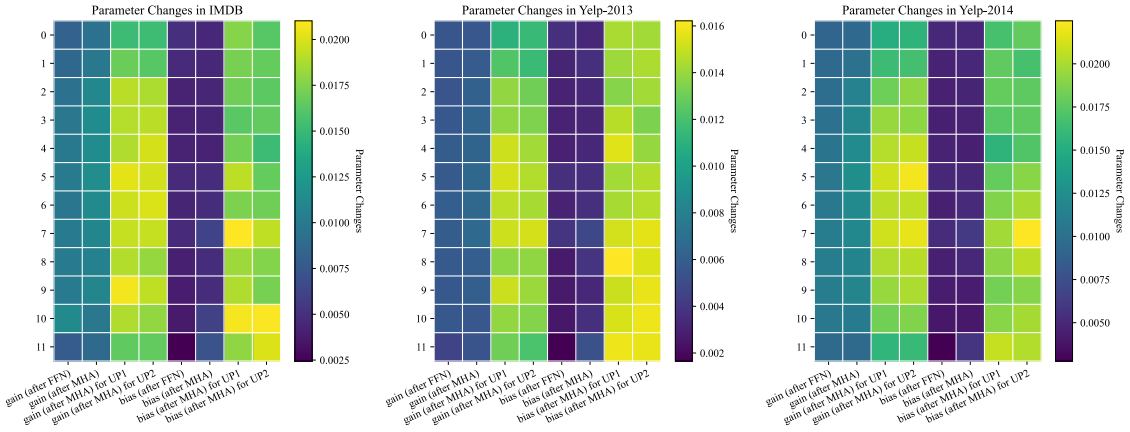


Figure 4: Visualization of the change of terms on IMDB, Yelp-2013, and Yelp-2014 datasets.

$d_u \& d_p$	256	512	768	1024
256	57.4	57.8	58.2	58.0
512	57.6	58.1	**58.5**	58.0
768	57.4	57.9	58.0	57.8
1024	57.2	57.7	57.8	57.7

Table 6: Acc (%) performance on dev dataset of IMDB with various dimensionalities of users (column-wise direction) and products (row-wise direction) embeddings.

datasets provided predetermined data splits, including training, development (dev), and test sets. Further statistics of the datasets were presented in Table 5.

A.2. Experimental Settings

We mainly used well pretrained checkpoints of uncased BERT and RoBERTa in the base version[3] to initialize the E2Transformer structure, which had 12 layers with a dimensionality d_h of 768. In terms of background embedding, UP word embeddings were set as d_u of 768 and d_p of 512, respectively, associated to the best results for the dev sets in the grid search strategy. The size of the sliding window W was set to 512. In the fine-tuning phase, AdamW (Loshchilov and Hutter, 2019) was used as the optimizer, with a learning rate of 2e-5. The minibatch size was 16, and early stopping at three epochs was implemented to avoid overfitting. The experiments were conducted on two-way RTX 3090 GPU devices, and the code was implemented by PyTorch.

[3]https://huggingface.co/

B. Further Experiments

B.1. Dimensions of Background Information

In the knowledge base, personalized background information is primarily present in the UP embeddings (e^u and e^p). The dimensionalities (d_u and d_p) of these embeddings exhibit a corresponding capability. To investigate the effect of UP embeddings and the hyperparameter selection, we used a grid search strategy to study the dev performance of Acc scores with different dimensionalities on IMDB dataset, as presented in Table 6. Evidently, very small or very large dimensionalities of the user or product equally degraded the final performance of the model. A possible reason was that very small dimensionalities cannot fully embed rich personalized knowledge, whereas very large dimensionalities may contain redundant information, resulting in overfitting.

B.2. Effect of LN-Tuning

Setups. Since the proposed methods operated LNs, we further investigated the effect of LNT and its combinations with the previous PEFT methods. Specially, for sparse fine-tuning, we took MHA, FFN, and bias terms into account; for adapters, we applied LoRA that introduced low rank matrices of MHA for downstream task adaptations.

Results. Comparative results were reported in Table 7. Initially, we found that pure LNT (or LNT only)

Models		IMDB		Yelp-2013		Yelp-2014		#IN	#FT
		Acc	RMSE	Acc	RMSE	Acc	RMSE		
FFT	E2LN ($\mathcal{B}$)	58.4	1.050	70.4	0.586	71.4	0.571	0.02	100
	E2LN ($\mathcal{R}$)	**59.8**	0.972	71.9	**0.562**	**73.0**	**0.555**	0.01	100
PEFT	E2LN ($\mathcal{B}$) LNT	44.8	1.158	64.6	0.676	65.1	0.674	0.02	0.05
	E2LN ($\mathcal{R}$) LNT	48.9	1.119	68.0	0.625	68.4	0.605	0.01	0.04
	E2LN ($\mathcal{B}$) MHA + LNT	58.4	1.052	70.3	0.595	71.3	0.582	0.02	25.94
	E2LN ($\mathcal{R}$) MHA + LNT	**59.8**	0.959	**72.1**	**0.562**	**73.0**	0.556	0.01	22.78
	E2LN ($\mathcal{B}$) FFN + LNT	57.7	1.058	70.4	0.604	70.7	0.585	0.02	51.80
	E2LN ($\mathcal{R}$) FFN + LNT	58.8	**0.948**	71.7	0.584	72.4	0.562	0.01	45.50
	E2LN ($\mathcal{B}$) LoRA ($q, k, v\&o$) + LNT	58.3	1.068	69.8	0.605	71.0	0.587	2.17	2.15
	E2LN ($\mathcal{R}$) LoRA ($q, k, v\&o$) + LNT	58.2	0.991	71.4	0.578	72.1	0.568	1.91	1.90
	E2LN ($\mathcal{B}$) LoRA ($q\&v$) + LNT	57.1	1.134	68.6	0.631	70.6	0.591	1.09	1.12
	E2LN ($\mathcal{R}$) LoRA ($q\&v$) + LNT	57.2	1.023	70.4	0.590	71.2	0.572	0.96	0.98
	E2LN ($\mathcal{B}$) BitFit + LNT	46.3	1.326	65.1	0.670	64.9	0.676	0.02	0.13
	E2LN ($\mathcal{R}$) BitFit + LNT	49.8	1.124	68.6	0.611	68.6	0.604	0.01	0.11

Table 7: Results of comparison between LNT and several previous PEFT methods in a dynamic combination. #IN(%) counted external or injected parameters against the whole parameters of PLM backbones and #FT (%) revealed the statistics of trainable parameter ratios for each pair of UP (ignoring the number of UPs).

degraded the performance on all three datasets. A possible reason might be that although LN had shown its effectiveness in UP injections, fine-tuning LN alone was not enough for task adaptation in sentiment analysis. Evidentially, the results of PEFT groups proved that LNT provided a PEFT method to shorten the gaps between pretraining and fine-tuning phase in PLMs via dynamically combing with previous PEFT methods; consequently, it (especially for MHA+LNT) gained on-par personalized sentiment analysis performance than FFT. Furthermore, it can be found that 1) sparse fine-tuning of MHA (MHA+LNT) outperformed those of FFN (FFN+LNT) in our settings with fewer trainable parameters; 2) and matrix-based PEFT methods (LoRA+LNT) achieved better results than bias-based PEFT methods (BitFit+LNT). This may be because that the MHA module and the matrix terms in transformer were critical factors in shortening gaps between PLMs' pre-training in large common corpus and their fine-tuning phase for document-level sentiment analysis.

Visualizations. We further visualized the change of the fine-tuned terms at each layer for the better understanding of LNT and UP injection. Specially, following (Qi et al., 2022) and (Ben Zaken et al., 2022), we used $\|\mathbf{t}_o - \mathbf{t}_f\|_1 / \dim(\mathbf{t})$ to measure the change of terms, where $\mathbf{t}$ presented the terms that would be tuned during downstream task adaptations, between initialized values $\mathbf{t}_o$ and fine-tuned values $\mathbf{t}_f$. We conducted experiments on IMDB, Yelp-2013 and Yelp-2014 with E2LN ($\mathcal{R}$) LoRA ($q\&v$) + LNT, as illustrated in Figure 4.

It can be first observed that, in our settings, the terms of gain and bias slightly changed where the gain terms changed more than the bias terms. Although the terms of gain and bias might not perform well in downstream task adaptations (i.e., pure sentiment analysis), they were significantly changed when UP information was injected, indicating our PLN method was capable of enlarging solution spaces of LN for personalization. Mover, varying UP information changed the LN factors (both gain and bias) to different distributions, revealing the diversities of personal preferences in societies. We also found that, with the incorporation of UP information, the changes in higher layers showed relatively larger than those in lower layers. This phenomenon also explained the aforementioned findings where high-layer injections outperformed low-layer injections in Table 3.