

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

**You Zhang<sup>1</sup>, Jin Wang<sup>1\*</sup>, Liang-Chih Yu<sup>2\*</sup>, Dan Xu<sup>1</sup>, Xuejie Zhang<sup>1</sup>**

<sup>1</sup>School of Information Science and Engineering, Yunnan University, Yunnan, P.R.China

<sup>2</sup>Department of Information Management, Yuan Ze University, Taiwan

Contact: {yzhang0202, wangjin}@ynu.edu.cn, lcyu@saturn.yzu.edu.tw

*Proceedings of The 2024 Joint International Conference on Computational Linguistics,  
Language Resources and Evaluation*

## Motivation

- Compared with traditional text classification, personalized review sentiment classification requires an intelligent system to identify fine-grained polarities in document-level reviews.
- We focus on presenting and using personalized and structural information leveraged for effective and efficient personalized sentiment analysis of PLMs' adaptations.

## Challenges - 1

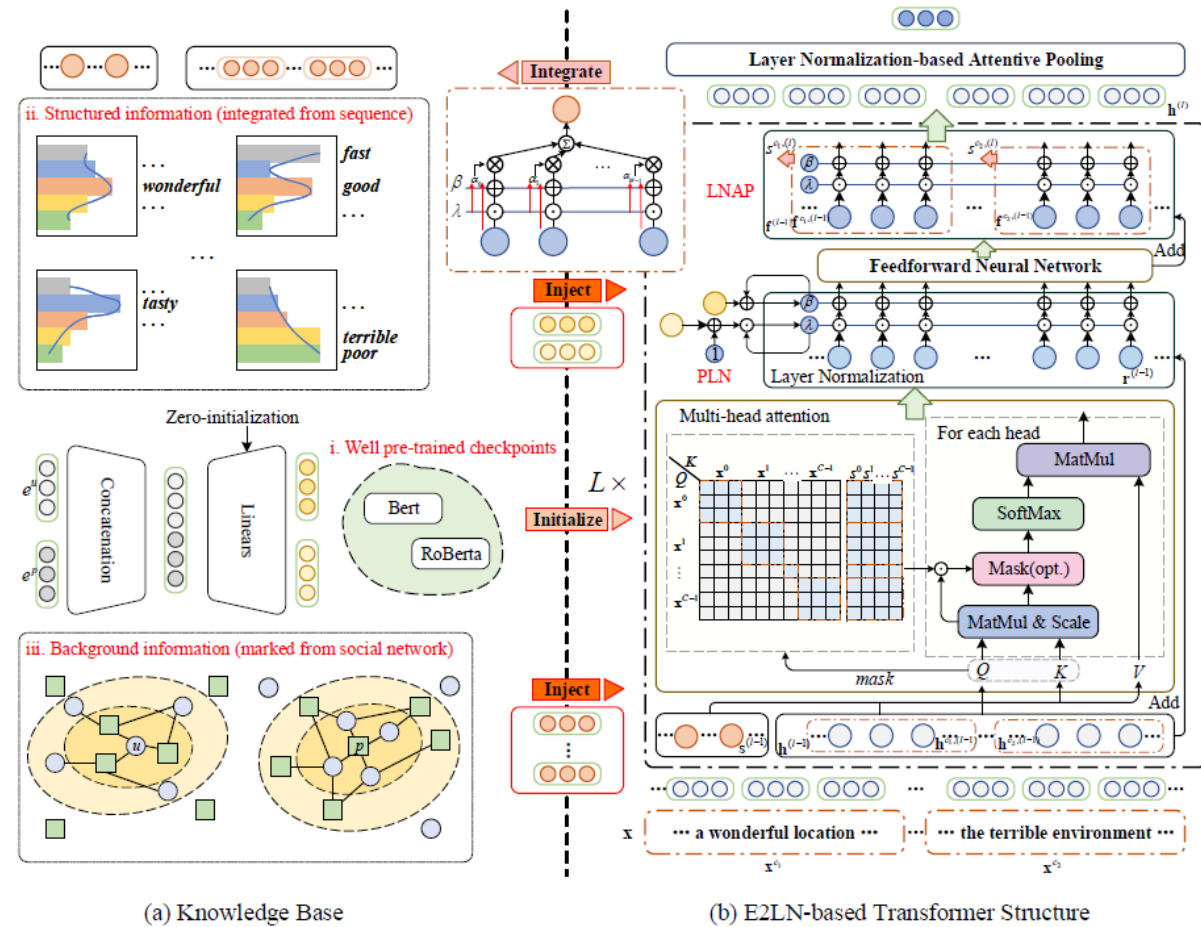
- The **computational complexity** in transformers increases at a quadratic rate with the input text length.
- Current solutions,
  - hierarchical approach
  - sparse attention matrix approach
- However,
  - neither can fully model the global context of documents and may have suboptimal performance in document-level review modeling tasks

## Challenges - 2

- The **heterogeneous mixes** of textual information from well-pretrained checkpoints and randomly initialized non-textual information make background information hard to inject directly into PLMs.
- Current solutions,
  - introducing different knowledge injection modules
- However,
  - sophisticated structure designs and large external parameters
  - fully model finetuning (FFT)

## Contribution

- We proposed a knowledge-enhanced and parameter-efficient layer normalization (**E2LN**) method for efficiently and effectively review modeling
- Regarding **knowledge enhancements**, a knowledge base was proposed for adapting PLMs, injecting background information, and extracting structured text information.
- Regarding **high parameter efficiency**, **LN-tuning**, external-parameter-free LN-based attentive pooling (**LNAP**), and weight-light personalized LN (**PLN**) were proposed.



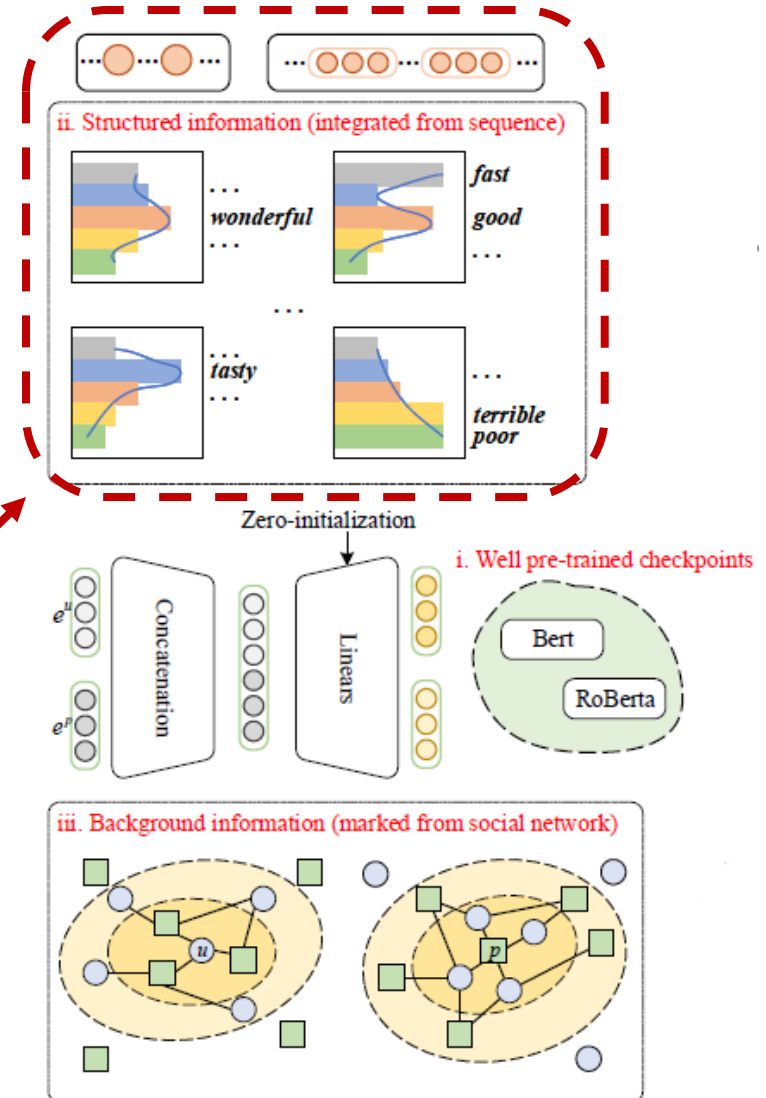
## Methodology-The Knowledge Base

- **Well-pretrained Checkpoints**
  - learned from a large number of general texts
  - fine-tuned for downstream tasks as general knowledge
- **Structural Text Information**
  - constructed from textual sequence representation via LNAP using a sliding window
  - stores global document contexts
- **Background Information**
  - To present discrete UP IDs via word embedding
  - injected in forward propagation
  - updated during backward propagation



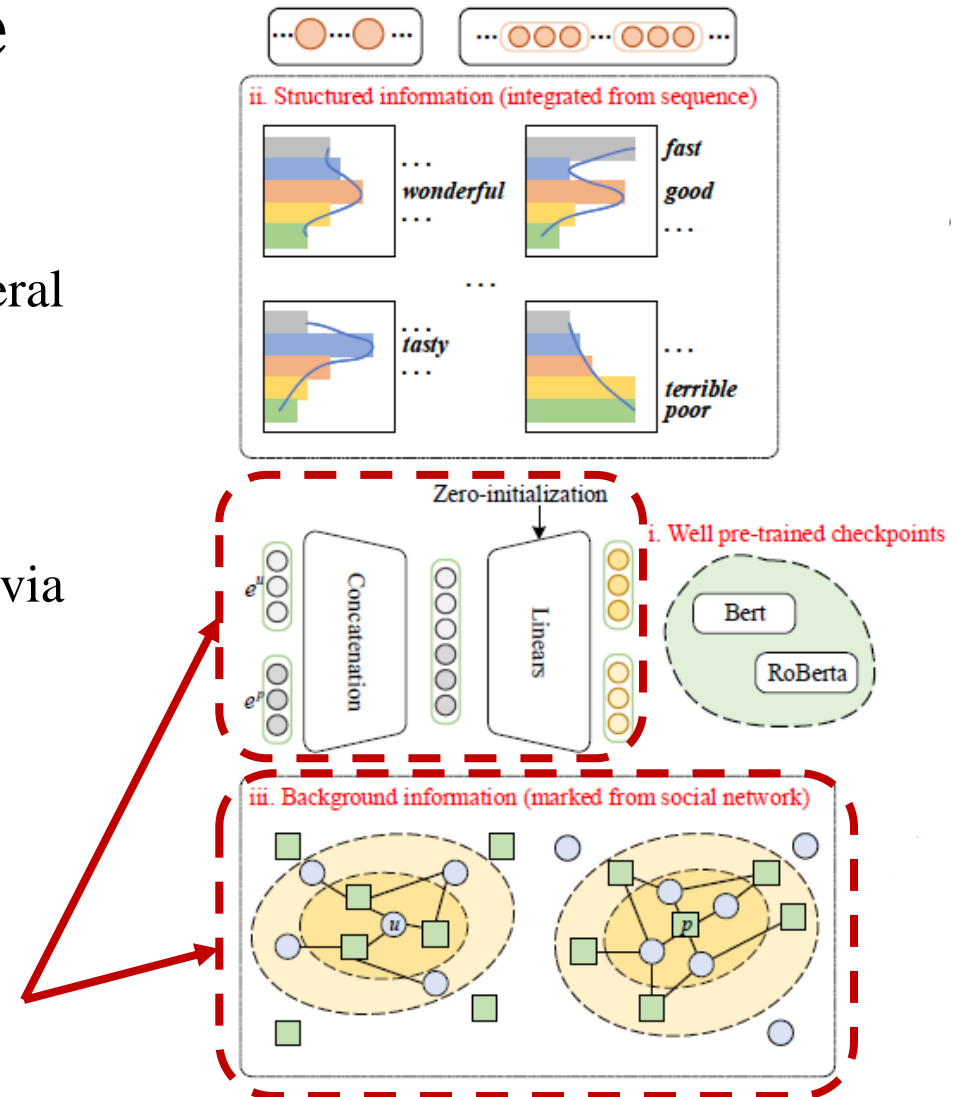
## Methodology-The Knowledge Base

- **Well-pretrained Checkpoints**
  - learned from a large number of general texts
  - fine-tuned for downstream tasks as general knowledge
- **Structural Text Information**
  - constructed from textual sequence representation via LNAP using a sliding window
  - stores global document contexts
- **Background Information**
  - To present discrete UP IDs via word embedding
  - injected in forward propagation
  - updated during backward propagation



## Methodology-The Knowledge Base

- **Well-pretrained Checkpoints**
  - learned from a large number of general texts
  - fine-tuned for downstream tasks as general knowledge
- **Structural Text Information**
  - constructed from textual sequence representation via LNAP using a sliding window
  - stores global document contexts
- **Background Information**
  - To present discrete UP IDs via word embedding
  - injected in forward propagation
  - updated during backward propagation



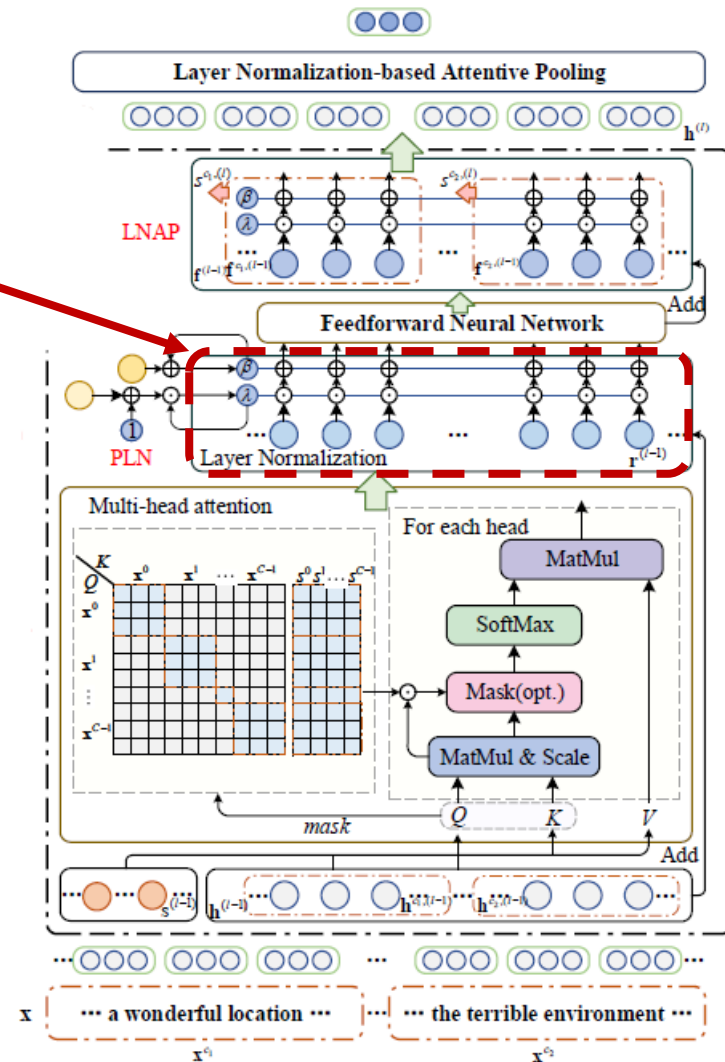
## Methodology-E2Transformer Structure

- LN
  - normalizes sequential representations
  - utilizes gain and bias term for power preservation

$$r'_n = \text{LN}(r_n; \lambda, \beta) = \frac{r_n - \mu_n}{\sigma_n} \odot \lambda + \beta$$

$$\mu_n = \frac{1}{d_h} \sum_{i=1}^{d_h} r_{ni}$$

$$\sigma_n = \sqrt{\frac{1}{d_h} \sum_{i=1}^{d_h} (r_{ni} - \mu_n)^2}$$



## Methodology-E2Transformer Structure

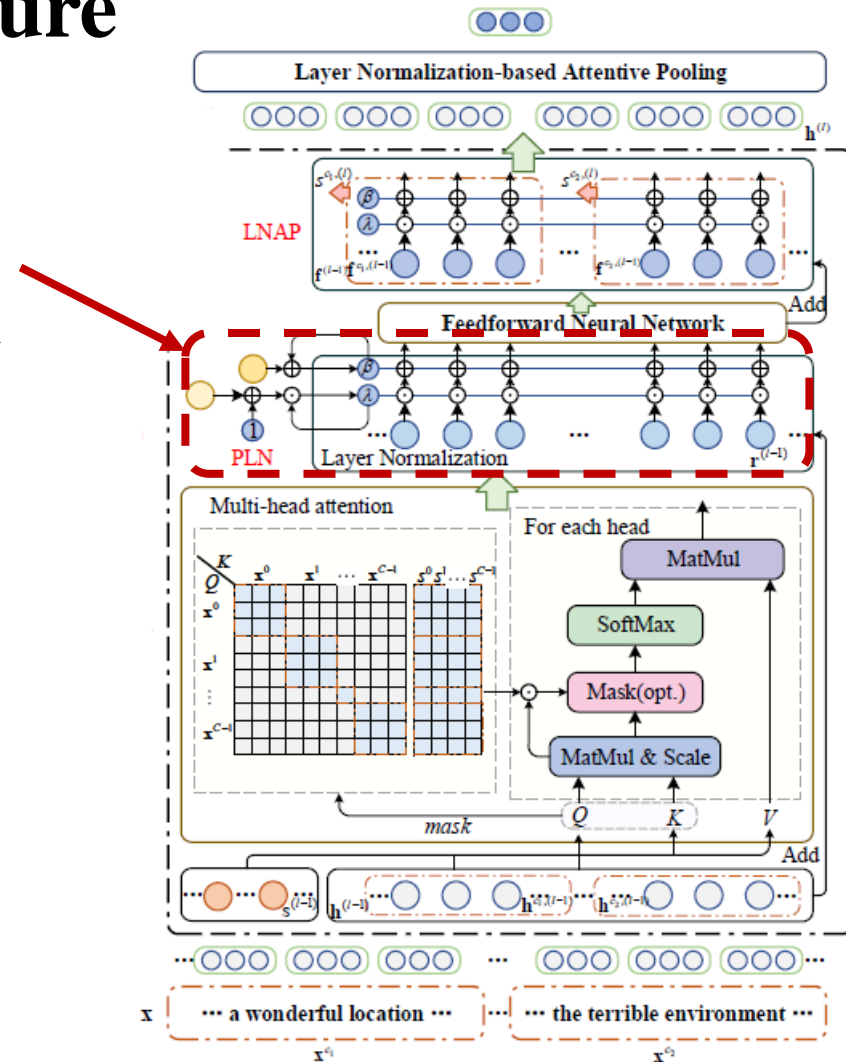
- **PLN**

- injects background information by personalizing the gain and bias parameters
- is more parameter-efficient due to vector-shaped parameters of gain and bias terms

$$r_n^{Pe} = \text{PLN}(r_n^{Se} + h_n, e^u, e^p) = \text{LN}(r_n^{Se} + h_n; \lambda^{Pe}, \beta^{Pe})$$

$$\lambda^{Pe} = (\mathbf{1} + \text{linear}_\lambda([e^u; e^p])) \odot \lambda$$

$$\beta^{Pe} = \text{linear}_\beta([e^u; e^p]) + \beta$$



## Methodology-E2Transformer Structure

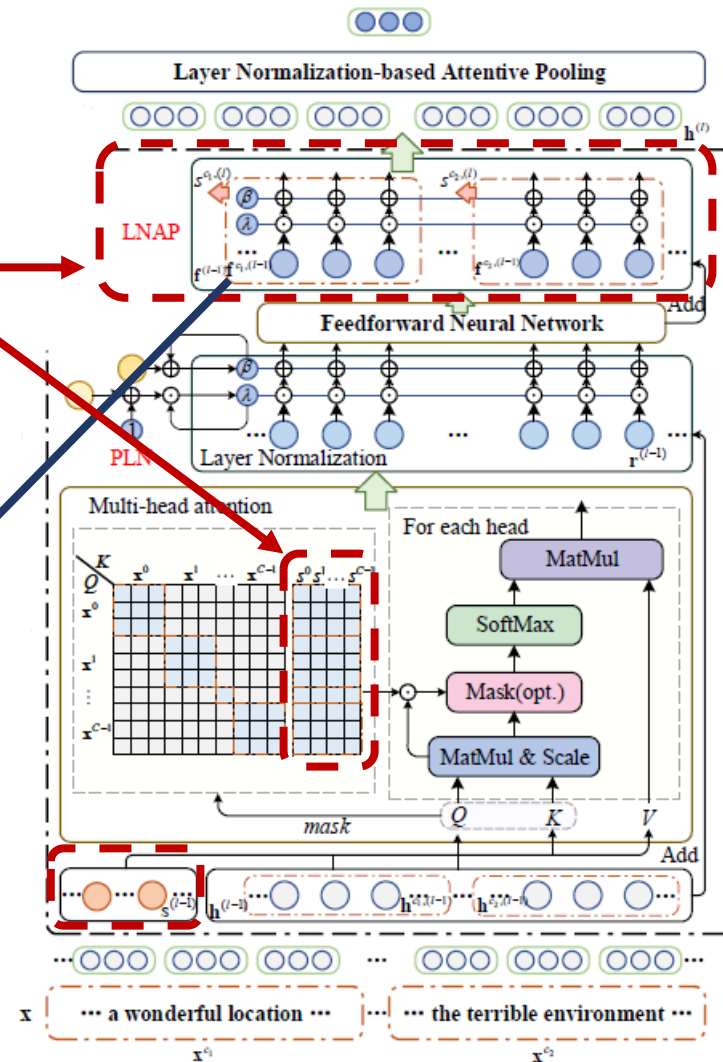
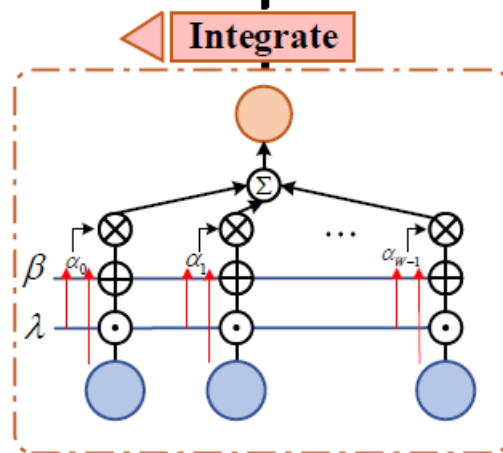
### • LNAP

- extracts structural text information based on gain parameters in LN which can reflect salience information in a representation vector
- is external-parameter-free
- Structural text information is injected into MHA for document representation

$$score_w^c = score(f_w^c) = \frac{f_w^c - \mu_w^c}{\sigma_w^c} \lambda^T \in R^W$$

$$\alpha_w^c = \frac{\exp(score_w^c)}{\sum_{w'=0}^{W-1} \exp(score_{w'}^c)} \odot mask(\mathbf{x}^c) \in R^W$$

$$s^c = \sum_w \alpha_w^c \cdot LN(f_w^c; \lambda, \beta) \in R^{d_h}$$



## Results on Comparative Results

- With the consideration of UP information, the performance of models was improved.
- With the consideration of global document information, the performance of models was improved.
- The proposed methods achieved the best performance by injecting structural text information and UP information.

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	<u>69.6</u>	<u>0.590</u>
	BigBird ( $\mathcal{R}$ )	<u>53.7</u>	<u>1.121</u>	<u>69.8</u>	<u>0.585</u>	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- $\mathcal{B}$ ) $\dagger$	57.2	1.050	70.0	0.593	71.4	0.587
	MAA (CK- $\mathcal{R}$ ) $\dagger$	58.3	1.015	71.6	<u>0.562</u>	72.5	0.567
	MAA ( $\mathcal{B}$ ) $\dagger$	53.0	1.141	69.3	0.594	70.0	0.579
	MAA ( $\mathcal{R}$ ) $\dagger$	54.8	1.074	71.5	0.578	72.4	<u>0.565</u>
Ours (FFT)	E2LN ( $\mathcal{B}$ )	58.4	1.050	70.4	0.586	71.4	0.571
	E2LN ( $\mathcal{R}$ )	<b>59.8</b>	<b>0.972</b>	71.9	<b>0.562</b>	<b>73.0</b>	<b>0.555</b>
+ 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	<b>59.8</b>	0.959	<b>72.1</b>	<b>0.562</b>	<b>73.0</b>	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.  $\dagger$  especially denoted performance reimplement from authors' original codes under the same experimental environments as ours.

## Results on Comparative Results

- With the consideration of UP information, the performance of models was improved.
- • With the consideration of global document information, the performance of models was improved.
- The proposed methods achieved the best performance by injecting structural text information and UP information.

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	<u>69.6</u>	<u>0.590</u>
	BigBird ( $\mathcal{R}$ )	<u>53.7</u>	<u>1.121</u>	<u>69.8</u>	<u>0.585</u>	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- $\mathcal{B}$ ) $\dagger$	57.2	1.050	70.0	0.593	71.4	0.587
	MAA (CK- $\mathcal{R}$ ) $\dagger$	58.3	1.015	71.6	<u>0.562</u>	<u>72.5</u>	0.567
	MAA ( $\mathcal{B}$ ) $\dagger$	53.0	1.141	69.3	0.594	70.0	0.579
MAA ( $\mathcal{R}$ ) $\dagger$	<u>54.8</u>	<u>1.074</u>	<u>71.5</u>	<u>0.578</u>	<u>72.4</u>	<u>0.565</u>	
Ours (FFT)	E2LN ( $\mathcal{B}$ )	58.4	1.050	70.4	0.586	71.4	0.571
	E2LN ( $\mathcal{R}$ )	<b>59.8</b>	<b>0.972</b>	71.9	<b>0.562</b>	<b>73.0</b>	<b>0.555</b>
+ 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	<b>59.8</b>	0.959	<b>72.1</b>	<b>0.562</b>	<b>73.0</b>	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.  $\dagger$  especially denoted performance reimplement from authors' original codes under the same experimental environments as ours.

## Results on Comparative Results

- With the consideration of UP information, the performance of models was improved.
- With the consideration of global document information, the performance of models was improved.
- • The proposed methods achieved the best performance by injecting structural text information and UP information.

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	<u>69.6</u>	<u>0.590</u>
	BigBird ( $\mathcal{R}$ )	<u>53.7</u>	<u>1.121</u>	<u>69.8</u>	<u>0.585</u>	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- $\mathcal{B}$ ) $\dagger$	57.2	1.050	70.0	0.593	71.4	0.587
	MAA (CK- $\mathcal{R}$ ) $\dagger$	58.3	1.015	71.6	<u>0.562</u>	<u>72.5</u>	0.567
	MAA ( $\mathcal{B}$ ) $\dagger$	53.0	1.141	69.3	0.594	70.0	0.579
	MAA ( $\mathcal{R}$ ) $\dagger$	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
	E2LN ( $\mathcal{R}$ )	<b>59.8</b>	<b>0.972</b>	71.9	<b>0.562</b>	<b>73.0</b>	<b>0.555</b>
+ 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	<b>59.8</b>	0.959	<b>72.1</b>	<b>0.562</b>	<b>73.0</b>	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.  $\dagger$  especially denoted performance reimplement from authors' original codes under the same experimental environments as ours.

## Analysis of Knowledge Enhancements – UP Injections

	Models	IMDB	Yelp-2013	Yelp-2014
	E2LN ( $\beta$ )			
<b>PLN</b>	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
<b>Module</b>	+ 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
<b>Layer</b>	1-6 layers only	57.4	70.2	70.7
	7-12 layers only	58.2	70.2	71.1

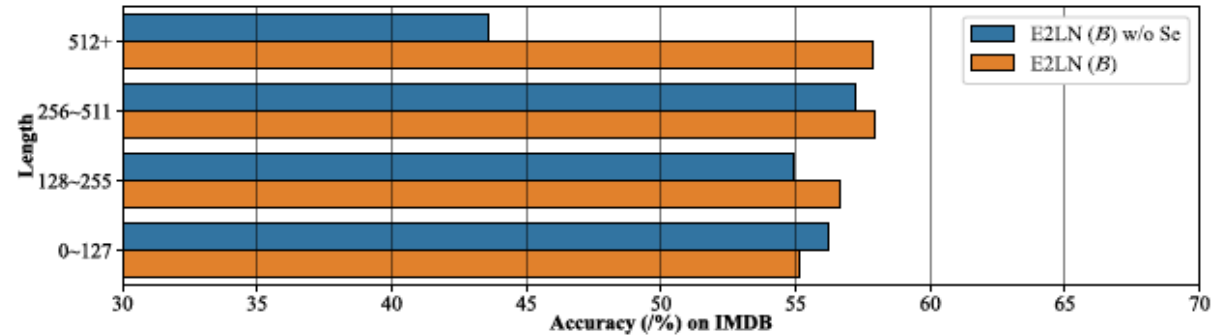
Table 3: Accuracy of E2LN ( $\beta$ ) for the investigation of UP injections. **PLN** means UP injections at different LNs. **Module** and **Layer** denote other modules (not matrix but only bias terms) and layers in the transformer structure activated for UP injection, respectively. w/ means only corresponding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

- Varying injection modules, different performance performed
- High-layer (7-12 layers) injections outperformed low-layer (1-6) injections

## Analysis of Knowledge Enhancements – LNAP

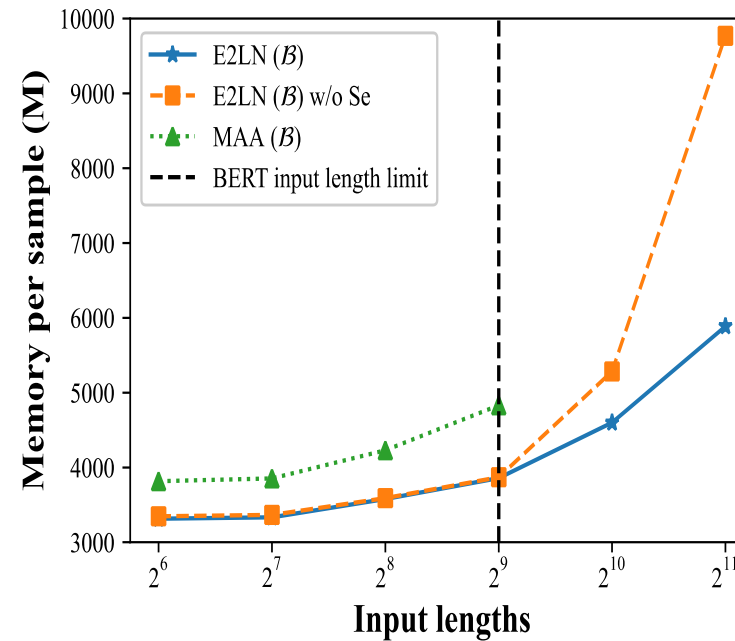
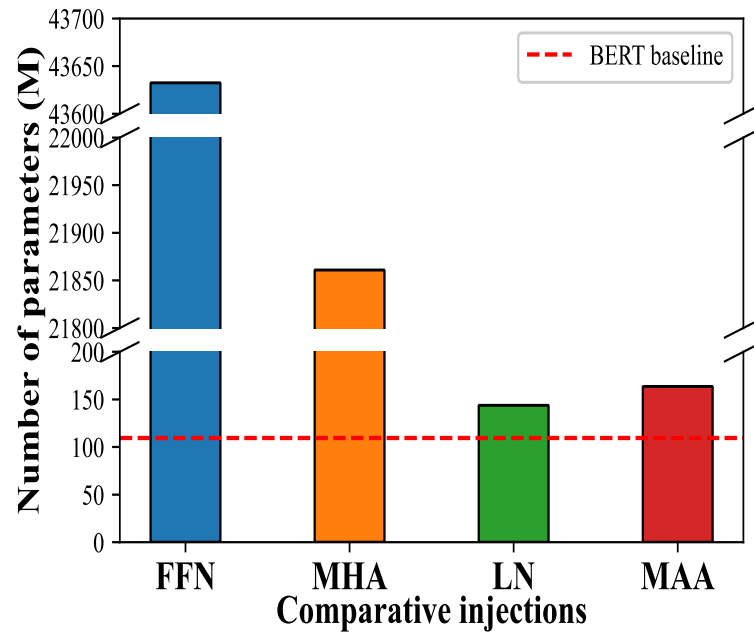
Models	IMDB	Yelp-2013	Yelp-2014
E2LN ( $\beta$ )			
+AvgP	57.8	69.7	71.1
+MaxP	53.1	66.8	68.6
+AttP	57.9	70.1	71.3
+LNAP	58.4	70.4	71.4

Table 4: Acc Performance (%) of various pooling methods embedded into E2LN ( $\beta$ ).



- LNAP performed the best and did not require additional modules with external parameters for fine-tuning
- LNAP could facilitate Transformer models to handle input reviews over 512 tokens, effectively.

## Analysis of Efficiency



- PLN requires much less parameters for personalized information injections
- LNAP could facilitate Transformer models to handle document-level reviews, efficiently.

## Summary

- **E2LN = (Knowledge-enhanced + Parameter-efficient) \* LN**
- **E2LN = PLMs + PLN + LNAP**



<https://github.com/yoyo-yun/E2LN>

LREC-COLING  2024



**Thank You for Listening**