Instructing the Learning of Language Model with the Token Interpretation to Improve Language Understanding

Anonymous ACL submission

Abstract

Pretraining Language Models (PLMs) have successfully established new state-of-the-art on a wide variety of Natural Language Understanding (NLU) tasks. We find that token interpretations, which provide rich and beneficial semantics, can instruct the learning procedure of PLMs. Therefore, this paper proposes a novel Language Understanding Enhancement method with Token Interpretation (LUETI), which could (i) strengthen the token semantics of input text by complementing the hidden state of token with the corresponding token interpretation, and (ii) strengthen the semantics of token embedding by leveraging representative vector of token interpretation for predicting the corresponding token. On a wide range of NLU tasks, LUETI, which could be viewed as a plug-in unit of conventional native PLMs, further improves the performance of BERT and GLM. Meanwhile, LUETI has even successfully achieved the similar performance as BERT and GLM only on a smaller scale of pretraining data.

1 Introduction

Recently, Pretraining Language Models (PLMs), such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), LaMDA (Thoppilan et al., 2022), GLM (Du et al., 2022), InstructGPT (Ouyang et al., 2022) and QUEST (Malaviya et al., 2023), are generally pretrained on large-scale unlabeled text corpora, and have achieved optimal performance for a diverse array of downstream Natural Language Understanding (NLU) tasks.

Especially, conventional PLMs usually adopt the encoder or decoder architecture of transformer, and focus on learning the bidirectional or unidirectional attention of context, thus suiting the different NLU tasks. These lines of methods can generally grouped into three classes: the bidirectional attention methods, the unidirectional attention methods and the bidirectional-unidirectional mixed methods. Wherein, the bidirectional attention methods (such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), DeBERTa (He et al., 2021) and GLM (Du et al., 2022)), utilize the encoder architecture of transformer and autoencoding task to learn the bidirectional attention of context, obtaining good performance in the NLU tasks; (ii) the unidirectional attention methods (such as GPT (Radford et al., 2019), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020)), InstructGPT (Ouyang et al., 2022) adopt the decoder architecture and autoregressive task to capture the left-to-right attention of context, succeeding in the task of long-text generation; and (iii) the bidirectional-unidirectional mixed methods (such as (Lewis et al., 2020), (Raffel et al., 2020), (Bao et al., 2020)), exploit the overall transformer to learn the unidirectional and bidirectional attention of context, achieving better performance in the text summarization and response generation tasks.

However, most of these conventional methods focus on learning the contextual semantics of input text, unfortunately ignore two facts: (i) the hidden state of token, is usually computed only by input text, and only could learn vague and incomplete semantics of the token; and (ii) the semantic information of the token embedding matrix learned by existing methods is universally insufficient, however any input text needs to be converted into vectors to the model through the token embedding matrix. Therefore, inspired by the following two motivations, this paper proposes a novel **L**anguage **U**nderstanding **E**nhancement method with **T**oken **I**nterpretation (**LUETI**): (i) **Motivation#1**: The token interpretation can supply beneficial semantics into input text. E.g., as shown in Fig.1, the token interpretation of "lipstick" can convey extra the "colour" and "lip" semantics into input text. (ii) **Motivation#2**: The token interpretation can effectively strengthen the semantics of token embedding.

E.g., as shown in Fig.2, adding the interpretation of "lipstick" into embedding vector can improve the prediction accuracy of the masked token "lipstick". To the best of our knowledge, the proposed work is the *first* taste for leveraging token interpretation to enhance the understanding of Language Models.

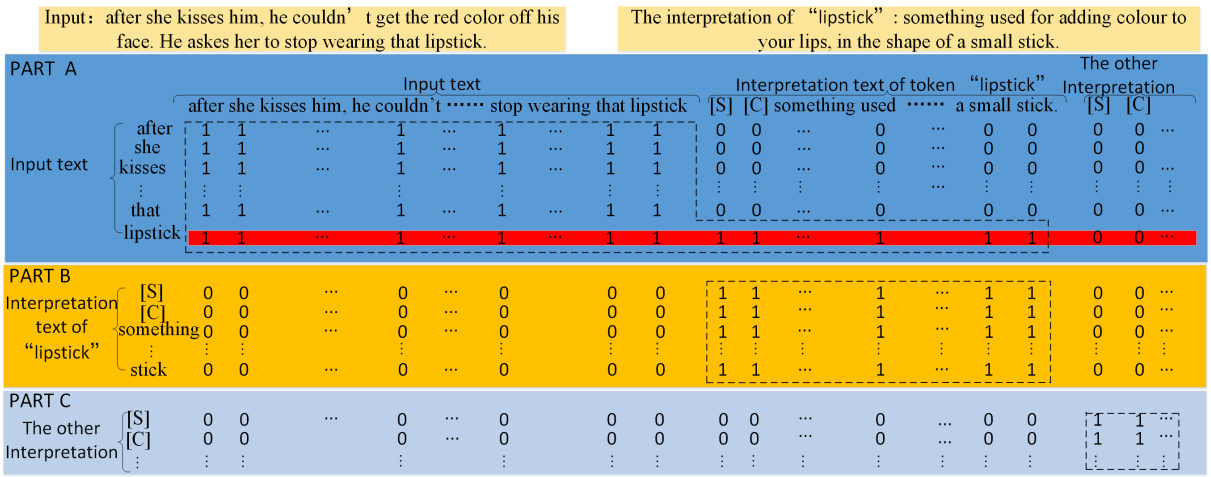
Summarily, we first append the token interpretation at the end of input text. For each token of initial input text, we compute its output state by the states of initial input text and the state of its corresponding interpretation. If the token from initial input text has no interpretation, its output is computed by the states of initial input text, following the conventional attention mechanism. Meanwhile, we compute the output state of token interpretation appeared in input text with its states, not communicating with the state of initial input text and the other token interpretation, and predict its representative vector as the corresponding token. Empirically, we show that on the SuperGLUE benchmark, the proposed LUETI, which is successfully applied to BERT and GLM models as a plug-in unit, substantially improves the performance than BERT and GLM by a large margin of 3.36% and 4.87% on average with the same amount of parameters and the same scale of pretraining data. Furthermore, we also demonstrate that LUETI achieves the similar performance with BERT and GLM models by a smaller scale of pretrained data.

2 Related Work

In effective PLMs currently presented, language model's pretraining procedure usually learns universal syntactic knowledge and semantic knowledge from large-scale unlabeled text corpora, and language model's finetuning procedure understand the task pattern and learn syntactic knowledge and semantic knowledge of target domain from labeled data. In this paper, we mainly focus on the masked language model and the related models (such as BERT and GLM), and discuss how to enhance their abilities of understanding language. The traditional masked language model, which is inspired by the Cloze task (Taylor, 1953), randomly masks some of the tokens from the input text, and the objective is to predict the original vocabulary ID of the masked token based only on its context. E.g., given a training corpus "He complained that after she kissed him, he couldn't get the red color off his face. He was annoyed and asked her to stop wearing that lipstick", a training instance $< x, y >$ can be obtained by masking the token "kissed", "color" and "lipstick". Especially, x is the sentence "He complained that after she [MASK] him, he couldn't get the red [MASK] off his face. He was annoyed and asked her to stop wearing that [MASK]"; and y is the text {kissed, color, lipstick}. The masked language model exploits the representation vector of x to predict the token in each [MASK] position. A good model must learn to encode syntactic (e.g., to predict "kissed") and semantic information (e.g., to predict the token "lipstick").

Existing pretraining models, which are almost pretrained on the transformer architecture and large-scale unlabeled text corpora, can generally grouped into three classes: the bidirectional attention methods, the unidirectional attention methods and bidirectional and unidirectional mixed models. The bidirectional attention methods, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), DeBERTa (He et al., 2021) and GLM (Du et al., 2022), base on the encoder architecture of transformer to perform the tasks of Masked LM, Next Sentence Prediction, blank infilling objective and so on. BERT (Devlin et al., 2019) captures the bidirectional attention of context, which does not contain the relations among the masked tokens. Further XLNet (Yang et al., 2019) and GLM (Du et al., 2022) respectively exploit the permutation language modeling and autoregressive blank infilling to capture the attention of the masked tokens with normal input tokens and already filled tokens. The unidirectional attention methods, such as GPT (Radford et al., 2019), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), base on the decoder architecture of transformer and autoregressive task to capture the unidirectional attention of context, thus learning left-to-right language models and succeeding in long-text generation. (Radford et al., 2019) and (Brown et al., 2020) exploit generative language models to complete NLU tasks such as question answering by directly predicting the correct answers without finetuning, given task instructions or a few labeled examples. (Ouyang et al., 2022) use reinforcement learning from human feedback to finetune GPT-3 to follow a broad class of written instructions. GPT-4 (OpenAI, 2023), which may have the similar architecture with GPT-3, adopts the transformer model pre-trained to predict the next token in a document.

The bidirectional and unidirectional mixed methods, such as (Lewis et al., 2020), (Raffel et al., 2020), (Bao et al., 2020), adopt bidirectional at-



(a) The adjacency matrix of initial input text and the corresponding interpretation text

Figure 1: The adjacency matrix of input text and the corresponding interpretation text: (i) The tokens from input text, not including the token "lipstick", can only attend to themselves (PART A, blue frame); (ii) The token "lipstick" can attend to input text and its interpretation word (PART A, red frame); (iii) The token interpretation can attend to themselves, not attending with the other interpretation and input text (PART B-C, orange frame and white frame).

tention for the encoder, unidirectional attention for the decoder, and cross attention between them, which is sucessful in conditional generation tasks, such as text summarization and response generation. BART (Lewis et al., 2020) feeds the input into the encoder and decoder to take the final hidden states of the decoder, which conducts NLU tasks. Instead, T5 (Raffel et al., 2020) exploits the conditional probability to formulate most language tasks in the text-to-text framework, thus training all tasks by a united mode. UniLM (Bao et al., 2020) unifies three pretraining models under the masked language modeling objective with different attention masks. However, most of these conventional methods focus on learning the contextual semantics of input text, unfortunately ignore two facts: (i) the hidden state of token is typically derived solely from the input text, which consequently leads to the acquisition of only nebulous and fragmentary semantics for the token; (ii) the semantic richness of the token embedding matrix acquired through current techniques is generally inadequate. Nonetheless, it is essential for any input text to be transformed into input vectors via this token embedding matrix for processing by the model.

3 LUETI Pretraining Framwork

This section proposes a novel language understanding enhancement method with token interpretation (LUETI), which helps to instruct the learning procedure of attention mechanism and token embedding by token interpretation (shown on Fig.1 and Fig.2).

3.1 Instructing the Learning Procedure of Attention Mechanism by Token Interpretation

Token interpretation is readily extracted from Longman Dictionary. E.g., the interpretation of token "lipstick" is "`something used for adding colour to your lips, in the shape of a small stick`", and the interpretation of token "complain" is "`to say that you are annoyed, not satisfied, or unhappy about something or someone`" etc. It should be noted that some tokens, which are not the real words, do not have the interpretation and some tokens have multiple semantics. This paper utilizes notation t to indicate a token, and utilizes notation e to indicate a text-formed interpretation respect to a certain token.

Given an input text x, $\mathcal{T}_x = \{t_1, \cdots, t_n\}$ denotes the token set tokenized by token vocabulary V, and then we generate a distilled token set $\mathcal{T'}_x$ by removing tokens without interpretations from $\mathcal{T}_x$. Furthermore, we intuitively obtain the token interpretation set $\mathcal{E'}_x$ respect to $\mathcal{T'}_x$. Wherein each element (i.e., e_i), which is tokenized by token vocabulary V, indicates the interpretation of token $t_i \in \mathcal{T'}_x$. With effort above, we cautiously append the token interpretation set $\mathcal{E'}_x$ at the end of input token text $\mathcal{T}_x$, to form the final input text.

In our LUETI, the token outputs of the final input text, are computed by following three cases:

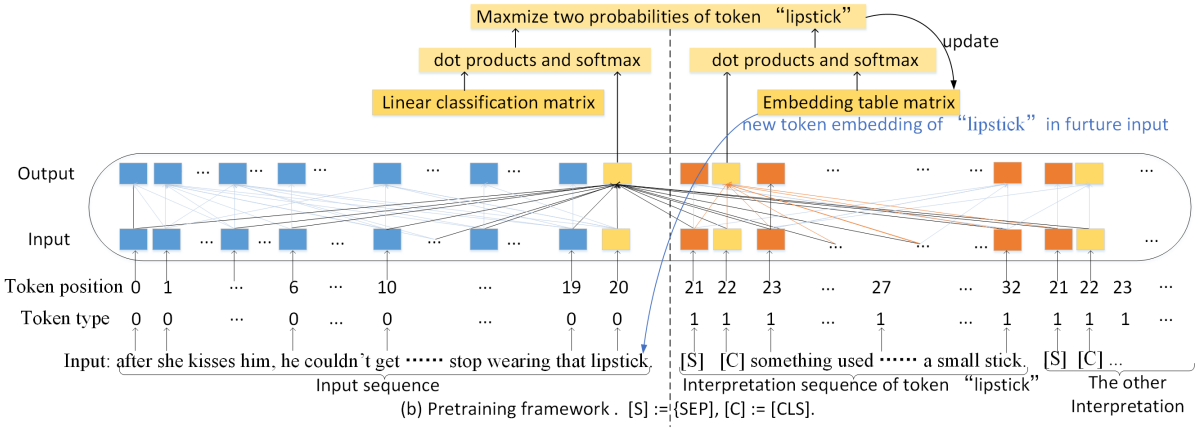


Figure 2: Pretraining framework. The hidden state of token "lipstick" is computed by the input text and its token interpretation. The CLS vector of "lipstick" interpretation is predicted as the token "lipstick".

- **Case#1: the output of token** $t_i \in \mathcal{T}_x \setminus \mathcal{T}'_x$, is only computed by the states of initial input text $\mathcal{T}_x$.
- **Case#2: the outputs of** $t_i \in \mathcal{T}'_x$, is computed by the states of input text $\mathcal{T}_x$ as well as its corresponding token interpretation $e_i \in \mathcal{E}'_x$, while not computed with the other token interpretation in $\mathcal{E}'_x \setminus \{e_i\}$.
- **Case#3: the output of interpretation** $e_i \in \mathcal{E}'_x$, is only computed by its states, while not computed with the states of the other token interpretation $\mathcal{E}'_x \setminus \{e_i\}$ and input text $\mathcal{T}_x$.

According the above three cases, we compute the adjacency matrix of final input text as follows. For input text x, its corresponding adjacency matrix $\boldsymbol{A}_x$ could be defined as follows:

$$\begin{aligned} \boldsymbol{F}_x &= [\mathbf{1}_n, \mathbf{0}_{m\times s}] \\ \boldsymbol{A}_x &= \boldsymbol{F}_x^\top \cdot \boldsymbol{F}_x \end{aligned} \tag{1}$$

Wherein, $n = |\mathcal{T}_x|$ represents the size of input token set in initial input text x, $m = |\mathcal{T}'_x|$ is the number of tokens with interpretation ($m < n$), and s indicates the predefined maximum length of interpretation. Besides, $\mathbf{1}_n$ denotes n-dimension vector where each element is set as 1, while $\mathbf{0}_{m\times s}$ denotes $(m \times s)$-dimension vector where each element is 0.

For each interpretation e_j in $\mathcal{E}'_x$, its adjacency matrix could be defined as follows:

$$\begin{aligned} \boldsymbol{F}_{e_j} &= [\textbf{one-hot}_{e_j}, \mathbf{0}_{e_1}, \cdots, \mathbf{1}_{e_j}, \cdots, \mathbf{0}_{e_m}], \\ \boldsymbol{A}_{e_j} &= \boldsymbol{F}_{e_j}^\top \cdot \boldsymbol{F}_{e_j} \end{aligned} \tag{2}$$

Wherein, $\textbf{one-hot}_{e_j}$ denotes a one-hot vector of length n with only the element of t_j's index in x being 1 and the rest being set as 0. Moreover, $\mathbf{1}_{e_j}$ represents a s-dimension vector with all the element set as 1, while $\mathbf{0}_{e_1}$ and $\mathbf{0}_{e_k}$ in the above formula are vectors with all the element being 0.

With efforts above, the overall adjacency matrix, could be defined as follows:

$$\boldsymbol{A} = \boldsymbol{A}_x + \sum_{e_j \in \mathcal{E}'_x} \boldsymbol{A}_{e_j}$$

Since the token $t_i \in \mathcal{T}'_x$ has been computed twice in equation (1) and (2), the main diagonal element of the token in $\boldsymbol{A}$ should be set to 1.

Following the conventional attention mechanism (Vaswani et al., 2017), we have a query $\boldsymbol{Q}$ and a pair of key-value $(\boldsymbol{K}, \boldsymbol{V})$ parameters at every layer. The weight, which is assigned to each value $\boldsymbol{V}$ to form the hidden states, is computed by follows:

$$\begin{aligned} Att(\boldsymbol{Q}, \boldsymbol{K}) &= softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}), \\ \boldsymbol{W} &= Att(\boldsymbol{Q}, \boldsymbol{K}) \odot \boldsymbol{A} \end{aligned} \tag{3}$$

Where the symbol $\odot$ represents the Hadamard product, and d_k is the dimension of vector $\boldsymbol{V}$.

By combining the $\boldsymbol{W}$ in equation (3) and values $\boldsymbol{V}$, we obtain the following output states of input:

$$\boldsymbol{H} = \boldsymbol{W} \cdot \boldsymbol{V}$$

3.2 Instructing the Learning Procedure of Token Embedding by Token Interpretation

3.3 Finetuning Procedure

This section exploits the token interpretation to instruct the learning procedure of token embedding,

System	COPA Acc.	RTE Acc.	BoolQ Acc.	WiC Acc.	CB Acc/F1-Mac.	MultiRC Em/Acc.	Average
LUETI performed on BERT with 110M ([B]) and 340M ([L]) parameters							
BERT$_{[B]}$	76.00	66.0650	73.2110	69.2790	75.0000/52.38	14.69/68.1518	61.8426
BERT$_{[B]+A}$	85.00	68.5921	73.8532	68.0251	80.3571/78.54	16.05/68.9356	67.4191
BERT$_{[B]+A+E}$	88.00	67.1480	73.4862	68.3386	78.5714/74.99	16.16/68.4818	66.8970
BERT$_{[L]}$	95.00	67.5090	73.2722	68.8088	80.3571/76.38	**16.47**/68.0487	68.2352
BERT$_{[L]+A}$	**97.00**	71.8412	**74.8318**	69.2790	**83.9286**/76.98	15.53/68.8119	69.7753
BERT$_{[L]+A+E}$	**97.00**	**72.9241**	74.0673	**69.4357**	**83.9286/82.84**	**16.47/68.9769**	**70.7053**
LUETI performed on GLM with 110M [B]) and 340M ([L]) parameters							
GLM$_{[B]}$	88.00	69.5921	73.4557	67.3981	75.0000/62.53	16.79/69.0594	65.2281
GLM$_{[B]+A}$	90.00	71.4801	73.4557	67.3981	80.3571/73.64	17.63/68.7500	67.8389
GLM$_{[B]+A+E}$	91.00	71.8412	73.4862	68.2962	82.1429/77.27	16.68/69.0182	68.7166
GLM$_{[L]}$	94.00	74.7293	76.3303	68.6520	89.2857/89.69	19.20/70.9983	72.8607
GLM$_{[L]+A}$	97.00	**76.1733**	76.9725	69.7492	89.2857/92.01	**22.14/72.1122**	74.4303
GLM$_{[L]+A_E}$	**99.00**	75.0936	**77.5229**	**70.6897**	**91.0714/93.20**	19.83/70.6064	**74.5958**

Table 1: Results of classification task on SuperGLUE. [ ]$_{+A}$ represents our variant instructing the attention by token interpretation, and [ ]$_{+A+E}$ represents our variant instructing the attention and token embedding by token interpretation.

which inspired by the task of Next Sentence Prediction (NSP) in conventional BERT (Devlin et al., 2019). Especially, as sketched in Fig.2, for each token interpretation e_j, we have appended the [CLS] token at the head of it, and hence there exists a [CLS] vector $\boldsymbol{c}_j$ for each token interpretation e_j, which will be used to predict the corresponding tokens. E.g., $\cdots$ The prediction probability of the token $t_i \in \mathcal{T}'_x$, could be formulated as follows:

$$p(t_i|\boldsymbol{c}_i) = \frac{\exp(M(i,i'))}{\sum_{k=1}^{|V|}\exp(M(i,k))}$$

$$\boldsymbol{M} = [\boldsymbol{c}_1;\cdots;\boldsymbol{c}_m]^\top \cdot \boldsymbol{T}$$

Wherein, i' indicates the index of token t_i in token vocabulary V. $\{\boldsymbol{c}_1,\cdots,\boldsymbol{c}_m\}$ represent the [CLS] vectors respect to the token interpretation in $\mathcal{E}'_x$, and operator $[\cdot;\cdot]$ indicates the connection operation. $\boldsymbol{T} \in \mathbb{R}^{d_h \times |V|}$ is the token embedding matrix, which will be updated in the pretraining procedure and then used to form the input embedding for token as well as token interpretation.

We follow the fine-tuning method of PET (Schick and Schütze, 2020) and GLM (Du et al., 2022). In detail, for a classification task, a function $f(\cdot)$ is used to transform the input text x into a sentence $f(x)$. Especially, $f(x)$ contains exactly one masked tokens or multiple masked tokens, i.e., the classification task can be viewed as a Cloze task. Furthermore, a projection function $g(\cdot)$ is defined to maps each label to a word or multiple words from vocabulary V. After obtaining the predicted tokens in the mask position of $f(x)$ by PLM, we map it to the corresponding labels by inverse function of $g(\cdot)$. E.g., we reformulate sentiment classification task as "TEXT. It is really [MASK]" and map the labels "positive" and "negative" as token "good" token "bad". Given the following input (x, y): (i) x is the text "after she kisses him, he couldn't get the red color off his face. He askes her to stop wearing that lipstick"; and (ii) y is the label of "negative".

Then, we apply $f(\cdot)$ to obtain an input representation $f(x)$: "after she kisses him, he couldn't get the red color off his face. He askes her to stop wearing that lipstick. It is really [MASK]". By PLM, we obtain the most likely token for the potential token in [MASK] position in $f(x)$ as "bad" not be "good", thus determining the label y as "negative". The conditional probability of predicting y given text x, is defined as as follows:

$$p(y|x) = \frac{p(g(y)|f(x))}{\sum_{y' \in Y} p(v(y)|f(x))}.$$

Where Y is the label set. Therefore the probability of the text being positive or negative, is equal to the prediction probability of token "good" or token "bad" in the masked task.

System	CNN/DailyMail			XSum		
	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
GLM$_{[116]}$	21.35	8.92	17.32	33.35	13.08	27.29
GLM$_{[116]+A}$	21.69	8.80	17.37	33.50	12.86	26.99
GLM$_{[116]+A+E}$	21.47	9.66	17.85	33.65	12.95	27.20
GLM$_{[232]}$	27.78	12.13	21.29	34.31	13.47	27.87
GLM$_{[232]+A}$	28.26	12.37	21.62	34.55	13.84	**28.79**
GLM$_{[232]+A+E}$	**28.48**	**12.42**	**21.74**	**35.18**	**14.17**	28.48

Table 2: Results of abstractive summarization task on the CNN/DailyMail and XSum test sets. $_{[116]}$ and $_{[232]}$ are respectively the abbreviation of 1160000 and 2320000 steps. All models are pretrained with 340M parameters.

4 Experiments

In this section, we view our LUETI as a plug-in unit, and perform it on two representative models BERT and GLM. We conduct comparative experiments on classification task, abstractive summarization task and question generation task, demonstrating the benefits of token interpretation for PLMs.

4.1 Pretraining Setup

To fairly evaluate the performance of our LUETI with BERT and GLM, we use BooksCorpus (Zhu et al., 2015) and English Wikipedia as our pretraining data, which are same as those used by BERT and GLM. Wherein, for Wikipedia, we extract text passages and ignore lists, tables, pictures, and references, etc., which has 13 GB plain text. Following BERT and GLM models, we also use the uncased wordpiece tokenizer with 30,522 vocabulary and 30,524 vocabulary[1] in our model pretraining.

Following our key motivations (shown in Figure 1 and Figure 2), we perform two facets of our LUETI on BERT and GLM with 110M and 340M parameters, and form 8 variant models, including BERT$_{[B]+A}$, BERT$_{[B]+A+E}$, BERT$_{[L]+A}$, BERT$_{[L]+A+E}$, GLM$_{[B]+A}$, GLM$_{[B]+A+E}$, GLM$_{[L]+A}$, GLM$_{[L]+A+E}$. Wherein, subscript $[B]$ and subscript $[L]$, which are respectively the abbreviation of $[Base]$ and $[Large]$, represent the 110M parameters (12 layers, hidden size 768, and 12 attention heads) and 340M parameters (24 layers, hidden size 1024, and 16 attention heads), respectively. The subscript $+A$, which is the abbreviation of $+Attention$, represents to instruct the learning of attention by the token interpretation (details in Sec. 3.1). The subscript $+A+E$, which is the abbreviation of $+Attention+Embedding$, represents to instruct the learning of attention and token embedding by the token interpretation (details in Sec. 3.2). To improve the effectiveness of GLM$_{[L]+A+E}$ and GLM$_{[B]+A+E}$, we add the task of Next Sentence Prediction of BERT to pretrain our LUETI.

[1]With the difference with initial GLM models, we only add the Sop and Eop tokens to form our GLM vocabulary V.

System	EM	F1	Average
GLM$_{[116]}$	61.6667	71.0236	66.3452
GLM$_{[116]+A}$	60.1786	69.8366	65.5076
GLM$_{[116]+A+E}$	63.0357	71.5030	67.2694
GLM$_{[232]}$	62.6190	71.2933	66.9562
GLM$_{[232]+A}$	63.0357	71.5234	67.2796
GLM$_{[232]+A+E}$	**64.4642**	**72.4133**	**68.4388**

Table 3: Results of question generation task on SQuAD. $_{[116]}$ and $_{[232]}$ are respectively the abbreviation of 1160000 and 2320000 steps. All models are pretrained with 340M parameters.

The input sequence of our LUETI contains the initial input text and its corresponding interpretation which is up to a maximum of 175 tokens in length (s=175). Due to insufficient hardware equipment, we set the max length of the pretraining text as 256 and the max number of corresponding interpretation as 4 (m=4). Thus the max sequence length of our LUETI is set as 956. Our schemes are pretrained with 2320,000 steps, batch size of 16 and sequence length of 956.

4.2 Classification Task

We conduct the experiments of classification tasks on the SuperGLUE (Wang et al., 2019). The SuperGLUE, which is the typical datasets to evaluate NLU systems, contains the subtasks of single-sentence classifications and sentence-pair classification. Following the ideas of PET (Schick and Schütze, 2020) and GLM (Du et al., 2022), we reformulate the classification tasks into Cloze questions and infill the masked tokens to predict the labels, which is accordance with the pretraining

System	COPA Acc.	RTE Acc.	BoolQ Acc.	WiC Acc.	CB Acc/F1-Mac.	MultiRC Em/Acc.	Average
$\text{GLM}_{[36]}$	87.00	**73.2852**	75.8410	66.7712	85.7143/82.32	20.67/70.3795	70.2477
$\text{GLM}_{[36]+A}$	**90.00**	**73.2852**	**76.9725**	**69.2790**	**87.5000/88.33**	**20.99/71.0190**	**72.2446**
$\text{GLM}_{[84]}$	**92.00**	73.6462	76.6972	**68.4953**	83.9286/81.50	**20.99**/70.9571	71.0268
$\text{GLM}_{[84]+A}$	91.00	**74.0072**	**77.2477**	68.0251	**87.5000/90.72**	20.67/**71.1015**	**72.5334**
$\text{GLM}_{[132]}$	92.00	**73.2852**	76.7890	68.8088	85.7143/87.02	20.36/71.1427	71.8900
$\text{GLM}_{[132]+A}$	**93.00**	**73.2852**	**77.1560**	**69.5925**	**89.2857/92.01**	**20.67/71.1634**	**73.2704**
$\text{GLM}_{[180]}$	**94.00**	74.7293	76.9419	69.5925	87.5000/87.76	21.51/71.1427	72.8971
$\text{GLM}_{[180]+A}$	**94.00**	75.0903	76.7278	70.2192	89.2857/92.01	19.62/71.0602	73.5017
$\text{GLM}_{[232]}$	94.00	74.7293	76.3303	69.2720	89.2857/89.69	20.15/70.9983	73.0570
$\text{GLM}_{[232]+A}$	**97.00**	**76.1733**	**77.5229**	**69.7492**	**89.2857/92.01**	**22.14/72.1122**	**74.4991**

Table 4: Ablation study over the token interpretation in instructing LUETI pretraining. Wherein, the subscript $_{[36]}$ is the abbreviation of 360,000 steps, etc. All models are pretrained with 340M parameters. $\text{GLM}_{[\ ]}$ is the initial schemes of GLM. $\text{GLM}_{[\ ]+A}$ is our schemes, which instruct the attention by token interpretation.

mode. The Cloze questions and other details can be found in PET and GLM.

With the same pretraining checkpoint in every class of finetuning experiments, we exploit the batch size of 18 and select the best fine-tuning learning rate (among 7e-6, 9e-6, 1e-5) on the Dev set. The results in Table 1 show that, with the same amount of training data and pretraining steps, our LUETI consistently improves the performance of GLM and BERT on all tasks with either base or large architecture. E.g., in the COPA, WiC and CB tasks, our $\text{GLM}_{[L]+A+E}$ outperform baseline $\text{GLM}_{[L]}$ by 5.00%, 2.04% and 3.51%. In the RTE and CB tasks, our $\text{BERT}_{[L]+A+E}$ outperform baseline $\text{BERT}_{[L]}$ by 5.42% and 6.46%. On average, as shown in 8th column of Table 1: (i) $\text{GLM}_{[L]+A+E}$ and $\text{GLM}_{[B]+A+E}$ outperforms $\text{GLM}_{[L]}$ and $\text{GLM}_{[B]}$ by 1.73% and 3.49%; and (ii) $\text{BERT}_{[L]+A+E}$ and $\text{BERT}_{[B]+A+E}$ outperforms $\text{BERT}_{[L]}$ and $\text{BERT}_{[B]}$ by 2.47% and 5.06%. It clearly demonstrate the advantage of LUETI in classification tasks. Meanwhile, further instructing the learning of token embedding by the token interpretation (notated as $_{[]+A+E}$), can also improve the ability of language understanding of PLMs. E.g., in the COPA, WiC and CB tasks, $\text{GLM}_{[L]+A+E}$ outperforms the $\text{GLM}_{[L]+A}$ by 2.00%, 0.94%, 1.79%, and 1.19%. All results demonstrate that LUETI can significantly improve the ability of language understanding of PLMs.

4.3 Abstractive Summarization Task

We evaluate the performance of LUETI in the abstractive summarization task with two datasets CNN/DailyMail (Nallapati et al., 2016) and XSum (Narayan et al., 2018). CNN/DailyMail, which aims to answer the fill-in-the-blank question, contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average) where the articles are extracted form CNN and Daily Mail websites and the summaries are manually generated from the corresponding articles. XSum consists of single sentence summaries and BBC articles, which is prefaced with an introductory sentence which is professionally written by the article author. Considering the fact that GLM model has better performance than BERT models at the abstractive summarization task, we pretrain three models $\text{GLM}_{[L]}$, $\text{GLM}_{[L]+A}$ and $\text{GLM}_{[L]+A+E}$ in the scale of 1,160,000 and 2,320,000 train steps, respectively denoted as $\text{GLM}_{[116]}$, $\text{GLM}_{[116]+A}$, $\text{GLM}_{[116]+A+E}$, $\text{GLM}_{[232]}$, $\text{GLM}_{[232]+A}$, $\text{GLM}_{[232]+A+E}$. The results (in Table 5) show that LUETI slightly improve the performance of the baseline GLM and BERT. We conjecture this can be attributed to the low efficiency of a maximum of 256 position tokens in pretraining LUETI, BERT and GLM.

4.4 Question Generation Task

Question generation task is evaluated on SQuAD V1.1 (Rajpurkar et al., 2016) which consists of 100k crowd-sourced question/answer pairs. Every question always has a corresponding answer in the paragraph of context from Wikipedia. We also exploit the aforementioned 6 models $\text{GLM}_{[116]}$, $\text{GLM}_{[116]+A}$, $\text{GLM}_{[116]+A+E}$,

System	COPA Acc.	RTE Acc.	BoolQ Acc.	WiC Acc.	CB Acc/F1-Mac.	MultiRC Em/Acc.	Average
LUETI performed on GLM with 340M parameters and 2,320,000 pretraining steps							
GLM_{A+E_0}	95.00	**74.7292**	76.6361	68.9655	83.9286/82.84	18.57/70.5446	72.1518
GLM_{A+E_1}	96.00	72.9242	77.0948	**70.8464**	87.5000/90.62	**19.94/70.9158**	73.2304
GLM_{A+E_2}	96.00	73.2852	77.0642	69.2790	87.5000/90.62	19.10/70.7508	72.9499
GLM_{A+E_4}	97.00	73.6462	**77.2477**	69.9060	**91.0714/93.20**	19.83/70.6064	74.0634
GLM_{A+E_6}	**98.00**	73.6462	**77.2477**	70.6897	**91.0714/93.20**	19.83/70.6064	74.2862

Table 5: Ablation study over the token interpretation in instructing LUETI finetuning.

$\text{GLM}_{[232]}$, $\text{GLM}_{[232]+A}$, $\text{GLM}_{[232]+A+E}$ as our variant models. The results are shown in Table 3. We observe that $\text{GLM}_{[232]+A+E}$ achieves best performance among all the comparative pretraining models on the EM and F1 metrics. On average, as shown in 4th column of Table 3, $\text{GLM}_{[232]+A+E}$ outperforms $\text{GLM}_{[232]}$ by 1.48%.

4.5 Ablation Study

This section conducts ablation analysis over two facets, thus further understanding the importance of token interpretation in instructing the learning of attention and embedding table in PLMs.

4.5.1 Effect of the Token Interpretation in Instructing Model Pretraining

In this section, we pretrain the GLM and $\text{GLM}_{[L]+A}$ in the scale of 360,000, 840,000, 1,320,000, 1,800,000 and 2,320,000 train steps, respectively denoted as $\text{GLM}_{[36]}$, $\text{GLM}_{[84]}$, $\text{GLM}_{[132]}$, $\text{GLM}_{[180]}$, $\text{GLM}_{[232]}$, $\text{GLM}_{[36]+A}$, $\text{GLM}_{[84]+A}$, $\text{GLM}_{[132]+A}$, $\text{GLM}_{[180]+A}$ and $\text{GLM}_{[232]+A}$. The results are presented in Table 4. In all pretraining steps, our LUETI outperforms the corresponding conventional GLM models. E.g, $\text{GLM}_{[232]+A}$, $\text{GLM}_{[180]+A}$, $\text{GLM}_{[132]+A}$, $\text{GLM}_{[84]+A}$ and $\text{GLM}_{[32]+A}$ respectively outperforms $\text{GLM}_{[232]}$, $\text{GLM}_{[180]}$, $\text{GLM}_{[132]}$, $\text{GLM}_{[84]}$, and $\text{GLM}_{[36]}$ by 1.44%, 0.60%, 1.38%, 1.50% and 2.00% on average. It further demonstrates that, LUETI can improve the ability of language understanding. Moreover, it is interesting that, $\text{GLM}_{[132]+A}$ and $\text{GLM}_{[36]+A}$, which are pretrained in a smaller scale pretraining data, respectively achieve the similar performance of $\text{GLM}_{[232]}$ and $\text{GLM}_{[132]}$.

4.5.2 Effect of the Token Interpretation in Instructing Model Fine-tuning

We exploit the model $\text{GLM}_{[L]+A+E}$ (2,300,000 train steps $\times$ batch size 16) as base model and respectively append 1, 2, 4 and 6 number of token interpretation at the end of SuperGLUE tasks (i.e., $m \in \{1, 2, 4, 6\}$). Hence, the corresponding schemes are denoted as GLM_{A+E_1}, GLM_{A+E_2}, GLM_{A+E_4} and GLM_{A+E_6}. The results (shown on Table 5) present that: compared with trivial fine-tuning methods, our LUETI enhanced finetuning significantly improves the model performance. E.g., GLM_{A+E_6}, $\text{GLM}_{A_E_4}$, GLM_{A+E_2} and GLM_{A+E_1} outperforms GLM_{A+E_0} by 2.13%, 1.91%, 0.80% and 1.08% on average. Meanwhile, the performance of LUETI enhanced fine-tuning is approximately positive correlation with the number of token interpretation appended into classification tasks. E.g., GLM_{A+E_6} outperforms GLM_{A+E_4} by 0.22% on average, and GLM_{A+E_4} outperforms GLM_{A+E_2} by 1.11% on average.

5 Conclusion

LUETI, which is a general and flexible pretraining framework, exploits the token interpretation to instruct the learning procedure of attention mechanism and token embedding, thus improving the ability of language understanding. Especially, it appends the token interpretation into initial input sequence, and compute the output of tokens from input with help of both the corresponding interpretation and initial input sequence. Meanwhile, it leverages the representative vector of each token interpretation for predicting corresponding token, thus conveying the token semantics into token embedding. Empirically, the proposed LUETI, which has been successfully applied to BERT and GLM models as a plug-in unit, substantially improves the performance than BERT and GLM. Furthermore, we also demonstrate that this work achieves the similar performance with BERT and GLM models by a smaller scale of pretrained data.

6 Limitations

A limitation in our work is that polysemous words may bring an negative influence on improving the ability of language understanding. The ability of language understanding may decline when multiple interpretations of polysemous words are simultaneously appended into the input sequence. Our scheme essentially exploit the semantics of token interpretation to supply the information in input sequence. However, when multiple semantic of polysemous words is completely independently, many token interpretation, which is not in context of input sequence, will bring some negative influence to the semantics of input sequence. Meanwhile, the tokens, which is with multiple parts of speech, will influence the ability of language model in understanding the parts of speech.

Another limitation is that the CLS vector of token interpretation may not well represent the token semantics. In our scheme, we exploit the CLS vector to represent the semantics of token interpretation and convey the token semantics into token embedding by predicting the loss of the CLS vector with the embedding matrix. The CLS vector is a important factor of influencing the ability of language understanding. To improve the representative ability of CLS vector, we additionally add the task of the Next Sentence Prediction in Bert model. Although we obtain a better results than the comparative schemes, there exist a further discuss to find a better representative methods of token semantics.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 642–652.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT:pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the International Conference on Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 320–335.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the International Conference on Annual Meeting of the Association for Computational Linguistics (ACL)*, page 7871–7880.

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. *CoRR*, abs/1907.11692.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In *Proceedings of the International Conference on Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 14032–14047.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the International Conference on Computational Natural Language Learning (CoNLL)*, page 280–290.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1797–1807.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, page 255–269.

Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. journalism bulletin. page 415–433.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, page 3261–3275.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet:generalized autoregressive pretraining for language understanding. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5754–5764.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 19–27.