DecBERT: Enhancing the Language Understanding of BERT with Causal Attention Masks

Ziyang Luo[1]*, **Yadong Xi**[2]*, **Jing Ma**[1], **Zhiwei Yang**[1,3],
Xiaoxi Mao[2], **Changjie Fan**[2], **Rongsheng Zhang**[2]
[1] Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China
[2] Fuxi AI Lab, NetEase Inc., Hangzhou, China
[3] Jilin University, Jilin, China
cszyluo@comp.hkbu.edu.hk, majing@hkbu.edu.hk
{xiyadong,zhangrongsheng}@corp.netease.com

Abstract

Since 2017, the Transformer-based models play critical roles in various downstream Natural Language Processing tasks. However, a common limitation of the attention mechanism utilized in Transformer Encoder is that it cannot automatically capture the information of word order, so explicit position embeddings are generally required to be fed into the target model. In contrast, Transformer Decoder with the causal attention masks is naturally sensitive to the word order. In this work, we focus on improving the position encoding ability of BERT with the causal attention masks. Furthermore, we propose a new pre-trained language model *DecBERT* and evaluate it on the GLUE benchmark. Experimental results show that (1) the causal attention mask is effective for BERT on the language understanding tasks; (2) our *DecBERT* model without position embeddings achieve comparable performance on the GLUE benchmark; and (3) our modification accelerates the pre-training process and *DecBERT w/ PE* achieves better overall performance than the baseline systems when pre-training with the same amount of computational resources.

1 Introduction

In recent years, Transformer model proposed by Vaswani et al. (2017) has supplanted the widely-used LSTM (Hochreiter and Schmidhuber, 1997) as an indispensable component of many NLP systems. There are two branches of model variant: Transformer Encoder and Transformer Decoder. The Encoder-based Language Models, e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), have achieved great success on many natural language understanding benchmarks (e.g. GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a)). The Decoder-based Language Models such as GPT-family (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020) have shown superior performances on natural language generation. All of them utilize the Multi-Head Self-Attention (MHA) mechanism (Vaswani et al., 2017). Since MHA is designed as an order-invariant mechanism (Lee et al., 2019), Transformer Encoder without the help of position embeddings should share the same intuitions with the bag-of-word model. On the other hand, in Transformer Decoder, the causal attention masks make the MHA different from that of the Transformer Encoder. Specifically, Tsai et al. (2019) have proved that MHA with such attention masks is not permutation equivalent, indicating that Transformer Decoder is sensitive to word order.

It is noticed that several studies focus on enriching the position information of BERT to improve the performance of natural language understanding (Dai et al., 2019; Dufter et al., 2020; He et al., 2020; Wu et al., 2021a; Ke et al., 2021), e.g., introducing extra learnable parameters to trace the word order. Previous analysis also indicate that the lower layers of BERT tend to capture rich surface-level language structural information such as position information (Jawahar et al., 2019). In this paper, to improve the language understanding of BERT, we propose to enrich the position information in the lower hidden layers instead of introducing extra learnable positional parameters.

To this end, we firstly design analysis experiments to examine the effectiveness of causal attention masks in terms of capturing position information. Then we propose a new pre-trained language model *DecBERT* by adding the causal attention masks into the lower layers of BERT (e.g., the first two layers) to enhance the position encoding ability. In this way, our proposed model is naturally sensitive to word order. Then we pre-train our *DecBERT* as a masked language model, following the same objective as BERT. To verify whether our modification can help BERT trace word order, we also make a comparison with a variant of our *DecBERT* that

* equal contribution

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1185 - 1197

excludes any position embeddings. The experimental results show that *DecBERT w/o PE* has 77 times (4.59 vs. 353.97) lower valid PPL score than BERT w/o PE and achieves comparable performance with BERT w/ PE on downstream tasks, corroborating the effectiveness of our modification. Furthermore, *DecBERT w/ PE* achieves better performances than BERT on most downstream tasks when pre-training with the same amount of time and computational resources. By analyzing the pre-training process, we find that our modification can also accelerate pre-training.

The contributions of this work are summarised as follows:

- We propose a novel pre-train model *DecBERT* utilizing the causal attention masks to enhance language understanding of BERT.
- We show that *DecBERT w/o PE* has comparable performance with BERT w/ PE, indicating that the causal attention masks are effective for modeling word order.
- When pre-training with the same amount of time and computational resources, *DecBERT w/ PE* achieves lower validation PPL and better overall performance on GLUE than BERT.

2 Background: Transformer

Transformer is a neural network model proposed by Vaswani et al. (2017), which relies on the multi-head self-attention (MHA) mechanism.

Input Layer. Due to the order-invariance of MHA (Lee et al., 2019), a token embedding is added with a position embedding as the input of Transformer Encoder or Decoder:

$$h_i = TE(x_i) + PE(i), \tag{1}$$

where x_i is a token at the i^{th} position. TE is a token embedding matrix and PE is a position embedding matrix. In the paper of Vaswani et al. (2017), they use a fixed sinusoidal PE:

$$\begin{aligned} PE[i, 2j] &= \sin(i/10000^{2j/d_m}), \\ PE[i, 2j+1] &= \cos(i/10000^{2j/d_m}), \end{aligned} \tag{2}$$

where j is the dimension and d_m is the model size. In the later work, Devlin et al. (2019) choose to use a learnable PE matrix.

Multi-head Self-attention (MHA). MHA takes a sequence of vectors $h = [h_1, h_2, ..., h_n]$ as input. Then they are transformed into three different vectors, query (Q), key (K) and value (V), by three linear transformations and passed to the multi-head self-attention (MHA). The computation process of a single head is:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3}$$

where d_k is the dimension of a single head. MHA repeats the same process for h heads. The outputs of all heads are concatenated together and passed through a linear projection W^O again:

$$\begin{aligned} H_i &= Attention(Q_i, K_i, V_i), \\ MHA(Q, K, V) &= concat(H_1, ..., H_h)W^O. \end{aligned} \tag{4}$$

Transformer Encoder and Decoder. An Encoder layer consists of multi-head attention following with a feed-forward network (FFN). The outputs of MHA and FFN are passed through a LayerNorm (Ba et al., 2016) with residual connections (He et al., 2016). Then we stack multi-layer to form a Transformer Encoder. The difference between Decoder and Encoder is that Decoder uses the causal attention masks to mask the attention values of the subsequent tokens so that Decoder can only decode tokens relying on the tokens in the past.[1]

3 Methodology

In this section, we first analyze the relationship between Transformer Decoder and position embeddings (section 3.1). Based on this analysis, we inject the causal attention masks into BERT to create our new pre-trained language models, *DecBERT* (section 3.2).

3.1 Transformer Decoder and Position Embeddings

Previous studies (Tsai et al., 2019) indicate that Transformer Decoder with causal attention masks is sensitive to word order. We wonder whether Transformer Decoder can perform well without position embeddings. We assume that if Transformer Decoder without any position

[1] We do not consider the Encoder-Decoder Seq2seq structure with cross attention here. Encoder and Decoder are used independently.

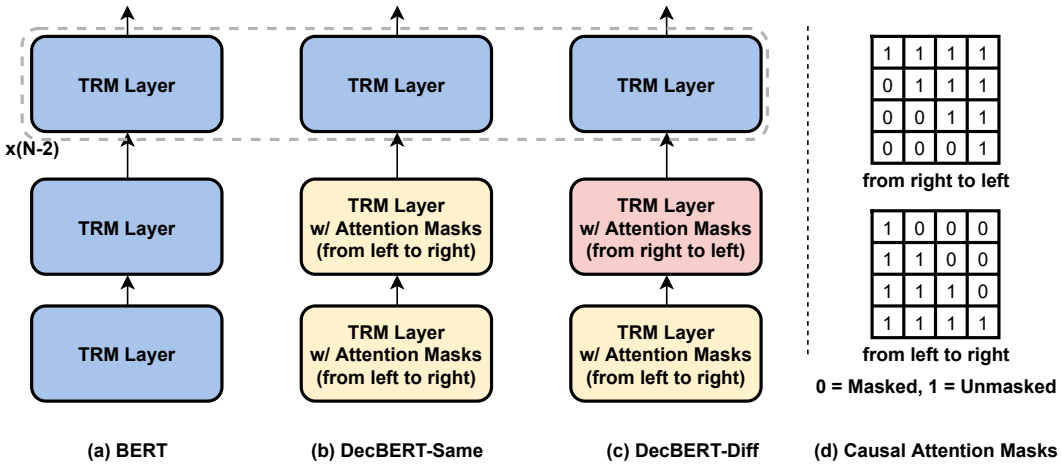

Figure 1: Model structures of BERT and DecBERT. TRM refers to the Transformer layer.

embeddings still retains comparable performance with its counterpart with position embeddings, it will corroborate that the causal attention masks are helpful for Transformer to encode word order. To this end, we design a straightforward experiment of causal language modeling respectively on English and Chinese data as followed.

Transformer Decoder	w/o *PE*	w/ *PE*
WikiText-103	23.52	23.37
Chinese Wiki	12.96	12.75

Table 1: Transformer Decoders perplexity (PPL) on WikiText-103 and Chinese Wikipedia validation sets. *PE* refers to the learnable position embeddings.

Basic Model. Our basic model is an 8-layer Transformer Decoder with 768 embedding size, 3072 feedforward layer hidden size, 12 attention heads and GELU activation function (Hendrycks and Gimpel, 2020), which is a smaller version of GPT and has 95M trainable parameters for English model and 77.5M for Chinese model.[2] We find that if we use a standard 12-layer GPT, the number of trainable parameters will be higher than the number of tokens in the WikiText-103 dataset. This has a risk to cause over-fitting, so we choose to use an 8-layer model.

Data and Training. We resort to two publicly available wikipedia datasets. The first one is the English WikiText-103 (Merity et al., 2017). We train and evaluate our language models on the standard splits of the WikiText-103, which contains 1.8M sentences for training and 3.76k sentences for evaluation. The second one is the Chinese Wikipedia which contains about 9.28M sentences. We randomly select 34k sentences for evaluation and 9.25M for training. We use Fairseq (Ott et al., 2019) to pre-process all the data into the binary files. All the English data is tokenized by SentencePiece tokenizer (Kudo and Richardson, 2018), which is the same as RoBERTa. All Chinese data is tokenized by character.

All models are trained with Fairseq. The training objective is the Causal Language Modeling objective. We use a batch size of 128 and train for 100k steps, optimized by Adam (Kingma and Ba, 2015). We also use the polynomial learning rate decay with 10k warmup steps. All models use the same hyper-parameters. We list the details in the Appendix. We use two NVIDIA A100 40GB GPUs to train each model. For the WikiText-103, it costs about 10 hours per model. For the Chinese Wikipedia, it costs about 8.5 hours per model.

Results and Discussion. Table 1 presents the perplexity (PPL) scores of Transformer Decoders with or without position embeddings on WikiText-103 and Chinese Wikipedia validation sets. Transformer Decoder w/o PE achieves comparable performance with its counterpart with learnable PE, which is only about 0.2 higher. This result reveals that the additional performance gain brought by position embeddings is small. Only relying on its causal attention masks, Transformer Decoder still can perform well. Combing our experiment and the previous studies (Tsai et al., 2019; Irie et al., 2019),

[2]The Chinese vocabulary size is smaller than English, so the Chinese model has fewer parameters.

the causal attention masks can make Transformer sensitive to word order.

3.2 Our DecBERT Model

In section 3.1, we conclude that Transformer with the causal attention masks is naturally sensitive to word order. Since the position information is inevitable for BERT, we propose to enhance existing BERT model based on causal attention masks.

In this paper, we intend to add the causal attention masks into all or some hidden layers of BERT. In this way, the specific layers with such masks are sensitive to word order by design, which can enhance the position encodings ability of BERT. Such framework can further result in better language understanding performances, e.g., in pre-trained language modeling, casual attention masks were added on all 12 layers of GPT (Radford and Narasimhan, 2018). However, comparing with BERT (Devlin et al., 2019), we observe that GPT lags behind BERT on almost all downstream tasks.[3] This is because self-attention mechanism with such masks only consider one-side information flow, it cannot process the input sentence comprehensively and has a high risk of language information loss. Therefore, we can conjecture that it is not effective to use the causal attention masks in all hidden layers. There is a strong need to maintain a balance between the gain of position encoding ability and the loss of language information.

In order to determine which layer(s) should add casual attention masks, we refer to the BERTology work (Jawahar et al., 2019) that conduct comprehensive experiments to analyze and interpret the information captured by each layer of BERT. The experimental results indicate that the lower layers of BERT capture rich language structure information. The position information is also a common structure information, so that we propose to add the causal attention masks into the lower layers (e.g., the first two layers[4]) to improve the position encoding ability of BERT. We denote our model as **DecBERT**. There are two versions of our model, **DecBERT-Same** and **DecBERT-Diff**. All of them are 12-layer base size models.

- **DecBERT-Same**: This model has a similar structure as BERT (see Figure 1(a)), but we use the causal attention masks to convert the first two Encoder layers into two Decoder layers with the same direction (from left to right). So the 12-layer model has 10 Encoder layers and 2 Decoder layers, which is shown in Figure 1(b). In this way, the first two layers are naturally sensitive to word order;

- **DecBERT-Diff**: This model is designed to enhance **DecBERT-Same** to gain more language information from different encoding directions. This model has a same structure as **DecBERT-Same**, except the second Decoder layer that has the opposite direction (from right to left). Figure 1(c) illustrates the model structure.

One would think that **DecBERT** is similar to Transformer with RNN layer (Neishi and Yoshinaga, 2019). Note that **DecBERT** is quite different from it, because **DecBERT** has similar structure as **BERT** and both of them require the same amount of computational time, which is much faster than that of Transformer with RNN.

4 Experiments and Results

4.1 Experimental Setup

Our experiments can be separated into two parts, small-scale pre-training scenario and large-scale pre-training scenario. Since the small-scale pre-training consumes much less time and fewer computational resources, we intend to answer several research questions in this part:

- Can DecBERT **without** any position embeddings still understand language well?

- Can DecBERT **with** position embeddings outperform BERT?

- Is using different directional causal attention masks more helpful than using the same directional?

- Why can DecBERT benefit from the causal attention masks, how do such masks affect the pre-training process?

For the large-scale pre-training scenario, we intend to examine whether the performance gap between our *DecBERT* and BERT will be diminished after scaling up the pre-training data size and time. Such settings can present a more comprehensive

[3] Although GPT and BERT are pre-trained with different objectives, the comparison is reasonable due to the same downstream tasks.

[4] We conducted massive experiments by adding the masks in the first, first-two, or first-three layer(s), and the first-two layers achieve the best performance.

view of whether our modification can benefit the pre-trained language models.

For a fair comparison, we re-implement BERT and pre-train it with the same settings as *DecBERT* in the small-scale and large-scale pre-training. We denote it as *BERT-reImp*.

Small-scale Pre-training Scenario. The pre-training data is the widely-used English Wikipedia Corpus. We randomly select 158.4M sentences for training and 50k sentences for validation. The pre-training objective is the Masked Language Modeling objective. We use a batch size of 256 and pre-train for 200k steps, optimized by Adam. All models use the same hyper-parameters. We list the details in the Appendix. We use four NVIDIA A100 40GB GPUs to pre-train each model, costing about 34.5 hours per model.

Large-scale Pre-training Scenario. Limited by time and computational resources, it is impossible for us to pre-train all models in the small-scale pre-training scenario from scratch in this setting. Thus, we decide to pre-train the best model in the small-scale scenario and the baseline model *BERT-reImp w/ PE* in this part. We use a large amount of pre-training data (around 160GiB[5]). The batch size is set to 4096 and the pre-training steps are 300k. We pre-train each model with 8 NVIDIA A100 40GB GPUs, costing about 15 days per model. The hyper-parameters details can be also seen in the Appendix.

Fine-tuning. To evaluate the language understanding ability of our models, we fine-tune them with 8 tasks of GLUE benchmark (Wang et al., 2019b), including SST-2 (Socher et al., 2013), QNLI (Rajpurkar et al., 2016), MNLI (Williams et al., 2018), QQP,[6] MRPC (Dolan and Brockett, 2005), CoLA (Warstadt et al., 2019), RTE[7] and STS-B (Cer et al., 2017). All fine-tuning hyper-parameters details are listed in the Appendix.

4.2 Small-scale Pre-training

Table 2 presents the pre-training perplexity scores of all systems on the validation set. Table 3 shows the performance of different systems on the GLUE benchmark. One can notice that our proposed models achieve lower valid PPL scores and higher overall scores on the downstream tasks.

Models	w/ *PE*	Valid PPL
Baseline		
BERT-reImp	False	353.97
BERT-reImp	True	4.28
Ours (w/o position embeddings)		
DecBERT-Same	False	4.59
DecBERT-Diff	False	4.59
Ours (w/ position embeddings)		
DecBERT-Same	True	4.12
DecBERT-Diff	True	**4.07**

Table 2: The validation set perplexity of all models in small-scale pre-training scenario. (w/ *PE* = with learnable position embeddings)

Can DecBERT without any position embeddings still understand language well? Since the self-attention of Transformer Encoder is order-invariant, the extra position information is inevitable for it to model language. Otherwise, it just becomes a bag-of-word model. From Table 2, we can find that the valid PPL score of *BERT-reImp w/o PE* is up to 353.97, which is about 82 times higher than its counterpart with position embeddings (4.28), revealing that this bag-of-word model cannot model language well. However, one can notice that *DecBERT* does not have such phenomenon. The valid PPL score of *DecBERT w/o PE* is only about 0.5 higher than *DecBERT w/ PE*. Compared with *BERT-reImp w/o PE*, the causal attention masks can decrease the PPL score by a large margin (from 353.97 to 4.59). After fine-tuning on downstream tasks, Table 2 indicates that *DecBERT-Same/Diff w/o PE* retains the same level performance as *BERT-reImp w/ PE*. These results reveal that *DecBERT* still can understand language well without the help of position embeddings, which is in line with our experimental results in section 3.1.

Can DecBERT with position embeddings outperform BERT? Table 2 shows that both *DecBERT-Same* and *DecBERT-Diff* have lower validation PPL scores than *BERT-reImp* (w/ PE). After fine-tuning on the downstream tasks, Table 3 reveals that they also have better performance on most tasks. These results confirm our belief that

[5]The details of our pre-training corpus can be seen in the Appendix.

[6]https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[7]https://aclweb.org/aclwiki/Recognizing_Textual_Entailment

Models	SST-2	QNLI	QQP	RTE	MNLI-m/mm	MRPC	STS-B	Avg.
Small-scale pretraining results on the dev sets								
BERT-reImp	89.56	89.24	90.14	64.40	80.14/80.62	**86.60**	86.22	83.37
Ours (w/o position embeddings)								
DecBERT-Same	89.58	89.50	90.16	62.68	79.56/80.42	85.88	**86.58**	83.05
DecBERT-Diff	90.30	88.86	90.28	59.28	79.78/80.78	86.08	86.06	82.68
Ours (w/ position embeddings)								
DecBERT-Same	90.12	89.18	**90.32**	64.78	80.48/80.64	86.24	86.34	83.51
DecBERT-Diff	**90.78**	**89.56**	90.08	**65.98**	**80.92/81.26**	85.86	86.24	**83.84**

Table 3: Different small-scale pre-training models' performance on the **dev sets** of GLUE benchmark. All results are averaged over five different random seeds (1, 2, 3, 4 and 5). MNLI-m is the matched version and MNLI-mm is the mismatched version. All tasks except STS-B use accuracy as their evaluation metrics. STS-B uses the Spearman rank correlation. The results are reported as $r \times 100$. **Bold** indicates the best score for each task.

Models	SST-2	QNLI	QQP	RTE	MNLI-m/mm	CoLA	MRPC	STS-B	Avg.
Large-scale pretraining results on the test sets									
BERT-reImp	**94.7**	91.5	**89.4**	66.5	85.9/85.1	56.3	85.4	**86.8**	82.4
DecBERT-Diff	94.5	**92.0**	89.3	**72.0**	**86.8/85.5**	**59.6**	**86.0**	**86.8**	**83.6**

Table 4: Different large-scale pre-training models' performance on the **test sets** of GLUE benchmark. All tasks except STS-B and CoLA use accuracy as their evaluation metrics. STS-B uses the Spearman rank correlation. CoLA uses the Matthews correlation coefficient. The results are reported as $r \times 100$. **Bold** indicates the best score of our models for each task.

our models can benefit from the causal attention masks. Such masks enhance the position encoding ability of BERT, leading to better language understanding ability.

Is using different directional causal attention masks helpful? The only difference between *DecBERT-Same* and *DecBERT-Diff* is that we adopt a different directional causal attention mask in the second layer. Table 2 shows that *DecBERT-Diff w/ PE* achieves the lowest validation PPL score (4.07). After fine-tuning on the downstream tasks, it also has the best overall score. These results confirm our belief that *DecBERT* can benefit from using different directional attention masks. Though the first two layers of *DecBERT-Diff* only consider one-side information flow, the model can learn to process different directional information in the first two layers. This design maintains a better balance between the gain of position encoding ability and the loss of language information.

Why can DecBERT benefit from the causal attention masks? The experimental results in the previous part indicate that the causal attention masks can increase the model's position encoding ability. Then such ability leads to better language understanding ability. However, the relation between these two abilities remains unclear. We analyze the pre-training process of our models to give a possible explanation.

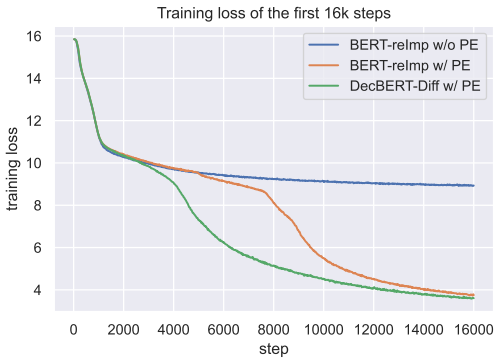

Figure 2: The pre-training loss of the first 16k steps. (Small-scale pre-training)

Our models' pre-training loss curves are presented in Figure 2 and 3. Since the randomly initialized Multi-head Self-Attention of BERT is a "balance" structure without any inductive bias, the model needs to learn suitable position embeddings to trace the word order during pre-training. In Figure 2, one can notice that the pre-training process of *BERT-reImp w/ PE* can be divided into four stages: (1) starting stage (0-1000 steps), (2) plateau

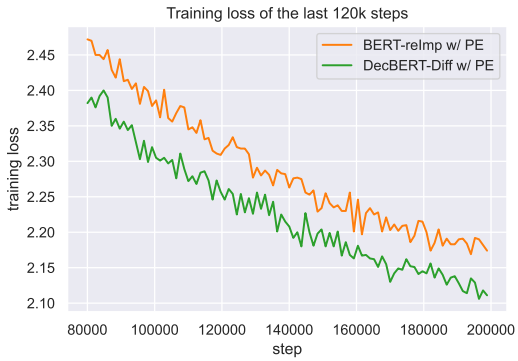

Figure 3: The pre-training loss of the last 120k steps. (Small-scale pre-training)

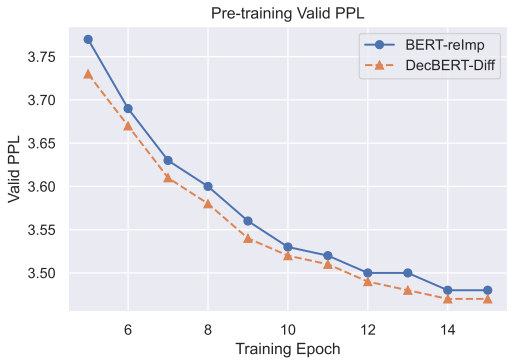

Figure 4: The PPL scores on validation set from epoch 5 to epoch 15 of our models. (Large-scale pre-training)

stage (1000-8000 steps), (3) "diving" stage (8000-10000 steps) and (4) convergence stage (10000-final steps). In the starting and plateau stages, *BERT-reImp w/ PE* has almost the same training loss as its counterpart without PE, which indicates that it is still a bag-of-word model and does not know how to make use of the position information. In the "diving" stage, the training loss of *BERT-reImp w/ PE* decreases rapidly, while *BERT-reImp w/o PE* starts to converge. This reveals that the word order information becomes more useful for models to understand language in such stage. In the convergence stage, the training loss decreases slowly to the end of the whole pre-training process.

So, how do the causal attention masks affect the pre-training process? The first two layers of *DecBERT* can break the "balance" of the multi-head self-attention by design. The position bias from the attention masks makes the first two layers sensitive to word order information. In Figure 2, one can notice that the plateau stage of *DecBERT* is shortened (from around 7000 to 3000 steps). This reveals that *DecBERT* does not need to spend as much time as BERT to learn to make use of the position information. It can escape from the bag-of-words sub-optimal point faster. Though the gap between *BERT-reImp w/ PE* and *DecBERT-Diff w/ PE* become smaller in the convergence stage, Figure 3 indicates that *DecBERT-Diff w/ PE* still has lower training loss in the whole pre-training process.

4.3 Large-scale Pre-training

In the large-scale pre-training scenario, we intend to verify whether our modification still achieves better performance. From Figure 4 and Table 4, one can find that the experimental results are similar to the small-scale pre-training scenario. For the validation PPL, *DecBERT-Diff* achieves lower scores than *BERT-reImp* in the whole pre-training process. Especially, at the 13^{th} epoch (265k steps), the valid PPL score of *DecBERT-Diff* is 3.48, which is the same as *BERT-reImp* at the 15^{th} epoch (300k steps). This suggests that the pre-training process of *DecBERT-Diff* is about 2 epochs faster than *BERT-reImp*. Combing our previous analysis, one advantage of our modification is that it can accelerate the pre-training process. Comparing the downstream tasks, one can also notice that the performance gap between *DecBERT-Diff* and *BERT-reImp* even becomes larger. The average score is 1.2 points higher.

All results in this part indicate that our modification is effective not only in the small-scale pre-training, but also in the large-scale pre-training. It can accelerate the pre-training process. When pre-training with the same amount of computational resources, our modification can achieve better performance on masked language modeling and downstream tasks.

4.4 Discussion

The analysis and experimental results detailed in the previous sections point out an interesting finding that the pre-training process of BERT can be divided into different stages. A similar phenomenon also can be found in the work of Kovaleva et al. (2021). In their work, they find that both scaling factors and biases of the Layer Normalization begin to diverge from their initialization values quickly in the "diving" stage. Especially, one/two specific neurons of the biases have larger and larger absolute values. Luo et al. (2021) indicates that such neurons are highly related to the positional informa-

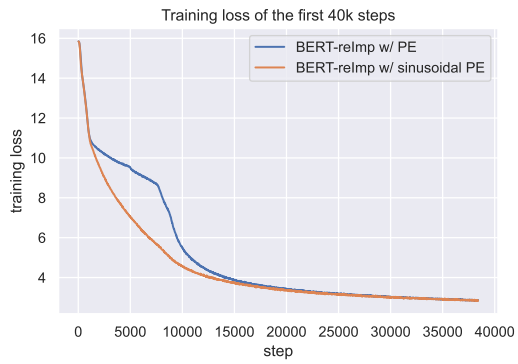

Figure 5: The pre-training loss of the first 40k steps. (Extra small-scale pre-training)

tion. These complement our possible explanation that in the plateau stage, the model needs to learn suitable position embeddings. Then in the "diving" stage, the model learns to adopt such embeddings to better model language. Our *DecBERT* models indicate that breaking the "balance" by design can help BERT better capture the position information, which leads to better performance.

One would wonder how about the fixed sinusoidal position embeddings. With such embeddings, BERT does not need to learn suitable position embeddings during pre-training. Based on our previous analysis, the plateau stage is possible to disappear. To examine whether such position embeddings are better, we conduct an extra small-scale pre-training experiment. The pre-training loss curve is in Figure 5, revealing that the plateau stage indeed disappears. This is in line with our previous results. However, in the convergence stage, we find that BERT with the sinusoidal PE has higher pre-training loss than using the learnable PE. This indicates that the learnable position embeddings are more suitable for BERT.

5 Related Work

The previous works (Vaswani et al., 2017; Shaw et al., 2018; Huang et al., 2019; Dai et al., 2019; Child et al., 2019) indicate that the self-attention mechanism of Transformer Encoder is permutation equivalent, so it needs to use the position embedding. Tsai et al. (2019) have proved that Decoder's self-attention is not permutation equivalent, indicating that Decoder is not a bag-of-word model as Encoder, but they do not conduct further analysis on Decoder's position encoding ability. Apart from the analysis, Irie et al. (2019) train the Transformer Language Models with speech dataset. They find that models without position embeddings have lower perplexity scores. Schlag et al. (2021a) introduce a new Linear Transformer Language Model with fast weight memories (Schmidhuber, 1992; Schlag et al., 2021b), which has lower perplexity without position encodings on the WikiText-103 dataset.

Furthermore, an explosion of work focuses on proposing a better method to add the position information into the pre-trained language model. Dufter et al. (2021) give a comprehensive introduction of different position encodings methods of Transformer. They divide position encodings into three approaches. One line of such work is to add position embeddings to the input before it is fed to the actual Transformer model (Vaswani et al., 2017; Shaw et al., 2018; Devlin et al., 2019; Kitaev et al., 2020; Liu et al., 2020; Press et al., 2020; Wang et al., 2020). The second line of work directly modify the attention matrix (Dai et al., 2019; Dufter et al., 2020; He et al., 2020; Wu et al., 2021a; Ke et al., 2021; Su et al., 2021). The last one combine the first two approaches together. However, all of them focus on introducing an extra set of parameters to trace the word order. Our work chooses to make use of the causal attention masks.

Most similar to our modification in Section 3.2, Im and Cho (2017) propose a self-attention based model which achieve better performance on SNLI task (Bowman et al., 2015) without the help of explicit position encodings. However, their models are different from the standard Transformer and use extra local attention masks to control the information flow. With the popularity of the Transformer model in the Computer Vision field, some works propose different methods to make Vision Transformer know word order implicitly (Chu et al., 2021; Yuan et al., 2021; Wu et al., 2021b), but all of them modify the models with convolution neural network (Lecun et al., 1998).

6 Conclusion

In this work, we introduce a new pre-trained model, called **DecBERT**, adopting the causal attention masks to enhance the language understanding of BERT. We conduct a series of experiments to verify the effectiveness of our models. Experimental results indicate that our proposed models achieve better performance than BERT on most downstream tasks when pre-training with the same amount of data and computational resources. Moreover, our

analysis also indicates that our models can accelerate the pre-training process.

7 Acknowledgments

We would like to thank Artur Kulmizev and the anonymous reviewers for their excellent feedback. This work is supported by the Key Research and Development Program of Zhejiang Province (No. 2022C01011), HKBU One-off Tier 2 Start-up Grant (Ref. RCOFSGT2/20-21/SCI/004) and HKBU direct grant (Ref. AIS 21-22/02).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. 2021. Conditional positional encodings for vision transformers.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2020. Increasing learning efficiency of self-attention networks through direct position interactions, learnable temperature, and convoluted attention. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3630–3636, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2021. Position information in transformers: An overview.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer. In *International Conference on Learning Representations*.

Jinbae Im and Sungzoon Cho. 2017. Distance-based self-attention network for natural language inference.

Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. Language modeling with deep transformers. *Interspeech 2019*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. Bert busters: Outlier dimensions that disrupt transformers.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit S. Dhillon, and Cho-Jui Hsieh. 2020. Learning to encode position for transformer with continuous dynamical model. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6327–6335. PMLR.

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.

Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. 2021. Positional artefacts propagate through masked language model embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5312–5327, Online. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Masato Neishi and Naoki Yoshinaga. 2019. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2020. Shortformer: Better language modeling using shorter inputs.

A. Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021a. Linear transformers are secretly fast weight memory systems.

Imanol Schlag, Tsendsuren Munkhdalai, and Jürgen Schmidhuber. 2021b. Learning associative inference using fast weight memory. In *International Conference on Learning Representations*.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4344–4353, Hong Kong, China. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020. Encoding word order in complex embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021a. Da-transformer: Distance-aware transformer.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021b. Cvt: Introducing convolutions to vision transformers.

Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. 2021. Incorporating convolution designs into visual transformers.

Yukun Zhu, Ryan Kiros, Richard 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.

A Hyper-parameters Details

Hyper-parameter	w/ or w/o PE
Number of Layers	8
Hidden size	768
FNN inner hidden size	3072
Attention Heads	12
Attention Head size	64
Dropout	0.1
Warmup Steps	10k
Max Steps	100k
Learning Rates	5e-5
Batch Size	128
Weight Decay	0.001
Learning Rate Decay	Polynomial
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.998
Gradient Clipping	0.1
Random Seed	1

Table 5: Hyper-parameters for pre-training the Transformer Decoder Causal Language Models.

Hyper-parameter	BERT/DecBERT
Number of Layers	12
Hidden size	768
FNN inner hidden size	3072
Attention Heads	12
Attention Head size	64
Dropout	0.1
Warmup Steps	24k
Max Steps	500k
Learning Rates	3e-4
Batch Size	4096
Weight Decay	0.01
Learning Rate Decay	Tri_stage
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.98
Gradient Clipping	2.0

Table 7: Hyper-parameters for pre-training the BERT and DecBERT (**large-scale pre-training**).

Hyper-parameter	BERT/DecBERT
Number of Layers	12
Hidden size	768
FNN inner hidden size	3072
Attention Heads	12
Attention Head size	64
Dropout	0.1
Warmup Steps	10k
Max Steps	200k
Learning Rates	1e-4
Batch Size	256
Weight Decay	0.01
Learning Rate Decay	Polynomial
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.98
Gradient Clipping	0.5
Random Seed	1

Table 6: Hyper-parameters for pre-training the BERT and DecBERT (**small-scale pre-training**).

B The details of the large-scale pre-training corpus

The first part is the same as BERT. We use the English wikipedia dump (about 17 GiB) and the bookcorpus (Zhu et al., 2015) (about 4 GiB). The second part is based on the Pile dataset (Gao et al., 2020), which is a large datasets with 800 GiB diverse text data. We randomly extract 64 GiB data from the Pile-cc block, 35 GiB data from the OpenWebText2 block and 43 GiB data from the Books3 block. The overall size of all data is about 163 GiB.

Hyper-parameter	MNLI	QNLI	QQP	RTE	SST-2	MRPC	STS-B	CoLA
Learning Rates	1e-5	1e-5	1e-5	2e-5	1e-5	{1e-5, 2e-5}	2e-5	1e-5
Weight Decay	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Batch Size	32	32	32	16	32	16	16	16
Warmup Steps	7432	1986	28318	122	1256	137	214	320
Max Steps	123873	33112	113272	2036	20935	2296	3598	5336
Adam ϵ	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6
Adam β_1	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
Gradient Clipping	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 8: Hyper-parameters for fine-tuning all models on downstream tasks. All models use the polynomial learning rate decay. Most of the hyper-parameters are recommended by Fairseq https://github.com/pytorch/fairseq/tree/main/examples/roberta/config/finetuning.