Learning Better Masking for Better Language Model Pre-training

Dongjie Yang[1,2], **Zhuosheng Zhang**[1,2,*], **Hai Zhao**[1,2,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
{djyang.tony,zhangzs}@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

Abstract

Masked Language Modeling (MLM) has been widely used as the denoising objective in pre-training language models (PrLMs). Existing PrLMs commonly adopt a Random-Token Masking strategy where a fixed masking ratio is applied and different contents are masked by an equal probability throughout the entire training. However, the model may receive a complicated impact from pre-training status, which changes accordingly as training time goes on. In this paper, we show that such time-invariant MLM settings on masking ratio and masked content are unlikely to deliver an optimal outcome, which motivates us to explore the influence of time-variant MLM settings. We propose two scheduled masking approaches that adaptively tune the masking ratio and masked content in different training stages, which improves the pre-training efficiency and effectiveness verified on the downstream tasks. Our work is a pioneer study on time-variant masking strategy on ratio and content and gives a better understanding of how masking ratio and masked content influence the MLM pre-training[1].

1 Introduction

Pre-trained language models (PrLMs) have played an essential role in many natural language processing tasks (Radford et al., 2018; Devlin et al., 2019; Bao et al., 2020; Guu et al., 2020; Yu et al., 2021; Zhang et al., 2022). Generally speaking, PrLMs can be seen as an automatic denoising encoder and may be conveniently obtained through a self-supervised learning way. Masked Language Modeling (MLM) pioneered by BERT (Devlin et al., 2019) is a widely used denoising method for language model pre-training (Lan et al., 2020; Clark et al., 2020). In MLM pre-training, a subset of tokens in a sequence is masked with a certain masking ratio, and the masked sequence is fed to the PrLM, which is required to predict the masked tokens.

Masking in MLM is a process in terms of sampling masked tokens from a huge data space to generate training batches, in which MLM may be heavily controlled by two main factors, masking ratio and masked contents. So far, only a few studies have ever considered optimal settings for better MLM from quite limited perspectives. Especially, all known works only take time-invariant MLM setting into account despite the huge time variance of the model during a lengthy pre-training. For example, carefully considered masked units like n-gram, entity and span (Sun et al., 2019; Joshi et al., 2020; Levine et al., 2021; Li and Zhao, 2021) are adopted throughout the entire pre-training. Another example is that exploring a good enough (but still fixed) masking ratio has also been considered in (Wettig et al., 2022). Given the circumstances that time-invariant masking applied in most MLM is not adaptive to the changeable process of language model pre-training, time-invariant setting hardly hopefully reaches an optimal outcome. This motivates us to explore the influence of masking ratio and masked content in MLM pre-training and propose time-variance MLM setting to verify our hypothesis for better PrLMs.

• **Masking Ratio**. Masking ratio controls the ratio between the number of tokens to predict and the left corrupted context. It determines the corruption degree that may affect the difficulty of restoring the masked tokens; that is, the larger the ratio is, the more masked contents model has to predict with less non-masked context. Our hypothesis is that at different training stages, the model may benefit from different masking ratios to balance the training from samples with different difficulties compared to the fixed ratio.

We first explore the influence of different mask-

*Corresponding author; This paper was partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

[1]https://github.com/mutonix/better_masking

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7255–7267

Table 1: Examples of masking function words and non-function words. Intuitively, it is much easier to predict the masked function words.

Masking Strategy	Example								
Function words	[MASK]	apple	[MASK]	day	keeps	[MASK]	doctor	[MASK]	.
Non-function words	An	[MASK]	a	[MASK]	[MASK]	the	[MASK]	away.	

ing ratios on downstream tasks at different stages throughout the entire pre-training instead of the only final stage. We find that a high masking ratio gives better performance for downstream tasks in early stage, while a low ratio has a faster training speed. Thus we choose a higher ratio as the starting point and decay the masking ratio to a lower value during the pre-training, namely Masking Ratio Decay (MRD), which can significantly outperform the performance of the fixed ratio. MRD indicates that MLM benefits from a time-variant masking ratio at different training stages.

• **Masked Content**. When placing all words with an equal and fixed opportunity throughout the entire pre-training for prediction learning, it may be unnecessary for some 'easy' words and insufficient for some 'difficult' words at the same time. Table 1 shows an intuitive example that the sequence with masked non-function words containing less information is much harder to predict compared to masked function words. Though in the very beginning, all words are unfamiliar to the models. As time goes on, the relative difficulties of words will vary when the pre-training status changes. We show that the losses of function words converge much faster than non-function words, which means non-function words are much harder for models to learn. Therefore, the high proportion of function words in a sequence leads to inefficiency if Random-Token Masking is applied.

To handle training maturity for different types of words, we propose POS-Tagging Weighted (PTW) Masking to adaptively adjust the masking probabilities of different types of words according to the current training state. PTW Masking makes the model have more chance to learn 'difficult' types of words and less chance for 'easy' ones from the perspective of part-of-speech. By introducing this adaptive schedule, our experimental results show that MLM benefits from learning mostly non-function words.

Our contributions are three folds: 1) We analyze the insufficiency of current masking strategies from the perspectives of masking ratio and masked content and give a better understanding of MLM pre-training in terms of masking. 2) To our best knowledge, this is a pioneer study to analyze the impact of time-variant masking both in masking ratio and masked content in MLM pre-training. 3) Our analysis shows that the time-variant masking schedules can significantly improve training efficiency and effectiveness. Our sources will be publicly available.

2 Preliminary Experiments

This section presents our preliminary experiments that motivate us to explore time-variant masking schedules. We train BERT-base (Devlin et al., 2019) with the widely-used English Wikipedia corpus to observe the influence of masking during pre-training by measuring the downstream performance on the SQuAD v1.1 dataset (Rajpurkar et al., 2016) (more experimental details will be given in Section 4). The experiments aim to study how the language model learns from the masked tokens when using conventional Random-Token Masking from the perspectives of the masking ratio and masked content.

2.1 Preliminaries of Masked Language Model

In general, Masked Language Modeling (MLM) is a denoising auto-encoding approach that is widely used in language model pre-training by reconstructing the corrupted sequences. To be specific, given a sequence $\boldsymbol{x} = \{x_i, x_2, \ldots, x_n\}$, we use a certain masking strategy $\boldsymbol{P}$ to replace $p\%$ tokens with special mask tokens. Accepting the corrupted sequence as input, a language model parameterized by θ is trained to predict the original tokens from masked ones in $\boldsymbol{x}$ using the pre-training objective stated below:

$$\mathcal{L}_{\mathrm{MLM}}(\boldsymbol{x}, \theta) = \mathbb{E}\left(-\sum_{i \in \mathbb{M}} log\, p_\theta(x_i \mid \hat{\boldsymbol{x}}) \right), \quad (1)$$

where $\hat{\boldsymbol{x}}$ is the reconstructed sequence that the language model samples from the hidden states,

and $\mathbb{M}$ denotes the index set of masked tokens where the loss will be calculated.

2.2 Masking Ratio: Influence of Pre-training Masking Ratio

The masking ratio determines the corruption degree of a whole sequence for model training. We first conduct a simple experiment to explore the impact of different masking ratios on downstream tasks at different training stages of entire pre-training.

We train BERT-base for 1M steps with a masking ratio of 15%, 25%, and 35% respectively, as shown in Figure 1a. During pre-training, checkpoints are saved every 50k steps, and finetuning is performed on the SQuAD v1.1 to observe changes in downstream performance. We find that the models using masking ratios of 25% and 35% have a gap of more than +1% F1 score in SQuAD compared with the model using a masking ratio of 15% at the beginning of training. However, in the second half training stage, the model with the masking ratio of 15% catches up with models with higher ratios in downstream performance.

2.3 Masked Content: Influence of Different Types of Words

In this section, we observe the influence of masked content by finding which kinds of words are more beneficial to pre-training. In terms of part-of-speech, words can be roughly divided into three categories: non-function words, function words, and the others (punctuations, symbols, etc.). If we mask all the function words and punctuations of a sentence, we can still infer roughly what the sentence is about. Instead, by masking all non-function words, we can hardly get any information from the sentence, as shown in Table 1.

To further explore the part-of-speech, we can speculate that, for the language model, masking different types of words leads to different difficulties for pre-training.

With the help of POS-tagging tools[2], we classify the words in the corpus into m categories[3] when doing pre-processing. In pre-training, for each type of words, we calculate the corresponding cumulative loss $\tilde{\ell}_{k,\,t}$ at t steps as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{MLM}}(\boldsymbol{x}, \theta) &= \sum_{k \in \mathcal{C}} \ell_{k,\,t}\ , \\ \tilde{\ell}_{k,\,t} &= \beta \cdot \tilde{\ell}_{k,\,t-1} + (1-\beta) \cdot \ell_{k,\,t}\ , \end{aligned} \tag{2}$$

where $k \in \mathcal{C}$ denotes the word type k in set $\mathcal{C}$ of m categories and $\beta \in (0, 1)$ is a coefficient to balance the exponential weighted average. We use exponential weighted average to smooth the losses because temporary losses of different batches vary greatly, leading to corresponding weights jittering (more details will be discussed in Section 3.3).

We train BERT-base for 200k steps with a fixed masking ratio of 15%. We record the cumulative losses of different types of words separately every 10 steps and observe the changes in losses. We find that the language model does have higher losses for masked non-function words and lower losses for masked function words. The latter quickly converges to very small values from the start, as shown in Figure 1b.

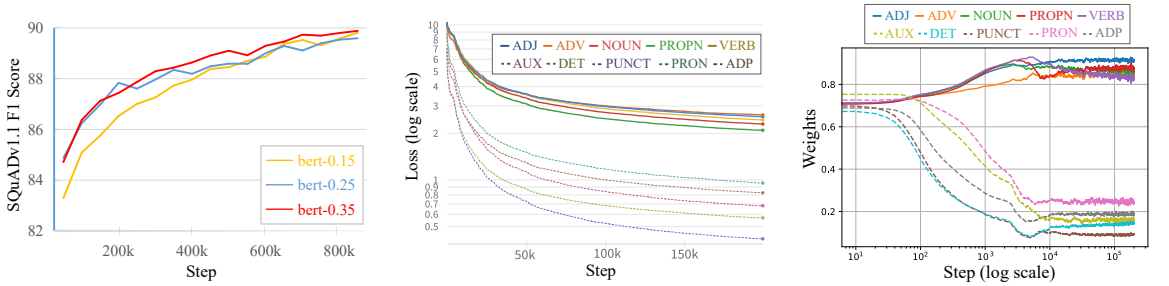
2.4 Analysis

Masking Ratio: Why Time-invariant Masking Ratio Is Not the Best Choice? From the experimental results in Figure 1a, there is such an empirical law: at the beginning, the downstream performance with a high masking ratio has a higher starting point but grows at a relatively slower speed and is caught up with the model with masking ratio of 15%. That is, the model with the masking ratio of 15% has a low starting point but boosts performance faster in the later stage. Given this observation, we show that we can apply a relatively high masking ratio to train models to get a better model using less time. On the other hand, we apply a lower masking ratio to train models, which obtains better downstream performance if we train for enough time. But if we use a decaying masking ratio instead of a fixed one, we can absorb the advantages of both high and low masking ratios.

Masked Content: Why Random-Token Masking Is Suboptimal? For a sentence, the numbers of non-function words and function words are quite similar. Therefore, for Random-Token Masking, the model pays equal attention to learning from these two kinds of words. However, the experimental results in Figure 1b show that the language model dissipates its effort to model some function words, of which losses have been very

[2]POS-tagging tool in spaCy: https://spacy.io/

[3]Part-of-speech classification from the Universal POS tag set: https://universaldependencies.org/u/pos/



(a) SQuAD v1.1 performance using different masking ratios.

(b) Cumulative losses of different types of words (shown 10 of them).

(c) Weighting changes of different types of words (shown 10 of them).

Figure 1: (a) Experiment on masking ratio: The evaluation metric F1 score measures the average textual overlap between the prediction and ground truth answer. (b) Experiment on masked content: Cumulative losses of different types of words (shown 10 of them). The losses of non-function words (in full line) are much greater than the losses of function words and punctuation (in dotted line). (c) Weight changes of PTW according to the cumulative losses in (b): Based on Equation 5, the masking weights of non-function words (in full line) are much higher than the weights of function words and punctuation (in dotted line).

low. Meanwhile, Random-Token Masking lets the model less likely learn those supposed-to-be-learn-more non-function words, which surely gives suboptimal pre-training consequences.

3 Time-variant Masking Strategies

In this section, we will present our exploration of time-variant masking on masking ratio and masked content inspired by our findings above. The overview of our time-variant masking is presented in Figure 3.

3.1 Masking Ratio Decay (MRD)

According to the observation in Section 2.2, we design an optimized Masking Ratio Decay (MRD) schedule. At the beginning of pre-training, we use a high masking ratio and decay the masking ratio using certain strategies, which is similar to learning rate decay without warmup. Assuming that the model generally adopts a fixed masking ratio $p\%$ for training, we use a very high masking ratio (about $2p\%$) as the starting point and a very low masking ratio (nearly zero) as the ending point in MRD.

3.1.1 Implementation of Two Decay Methods

We have tried two kinds of MRD to dynamically adjust the masking ratio, namely linear decay and cosine decay as follows:

$$\mathcal{M}_{linear}(t) = (1 - \frac{t}{T}) \cdot 2p\%, \quad (3)$$

$$\mathcal{M}_{cosine}(t) = (1 + cos(\frac{\pi}{T}t)) \cdot p\% + 0.02, \quad (4)$$

where $\mathcal{M}(t)$ is the current masking ratio at training step t and T is the total training step. Linear decay starts at $2p\%$ and decays to 0, while cosine decay starts at $2p\% + 0.02$ and decays to 0.02, as shown in Figure 2.

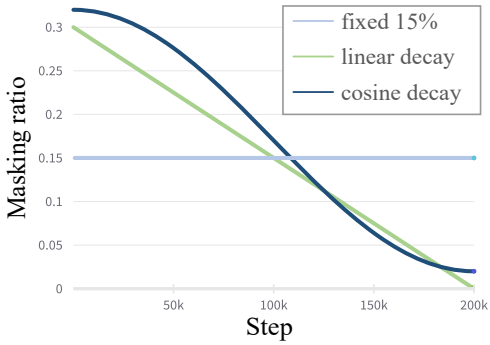


Figure 2: Masking Ratio Decay schedules ($p = 15$).

3.1.2 Details of Design Intention

We choose the starting point of $2p\%$ and ending point of 0 because the model using MRD can learn almost the same number of masked tokens as the baseline using a fixed masking ratio $p\%$ due to the central symmetry of linear and cosine functions for fair comparisons. The reason why we add 0.02 to the cosine decay is that the value of cosine function (masking ratio) is nearly 0 in the final 5% steps, which means there are no masked tokens for model to learn (and loss diminishes to 0). Thus we set a small number (0.02) to make the model keep training in the final stage.

3.2 Analysis for How MRD Works

MRD reminds us of the Simulated Annealing (SA) algorithm (Kirkpatrick et al., 1983), which is a greedy algorithm for optimization. In the SA

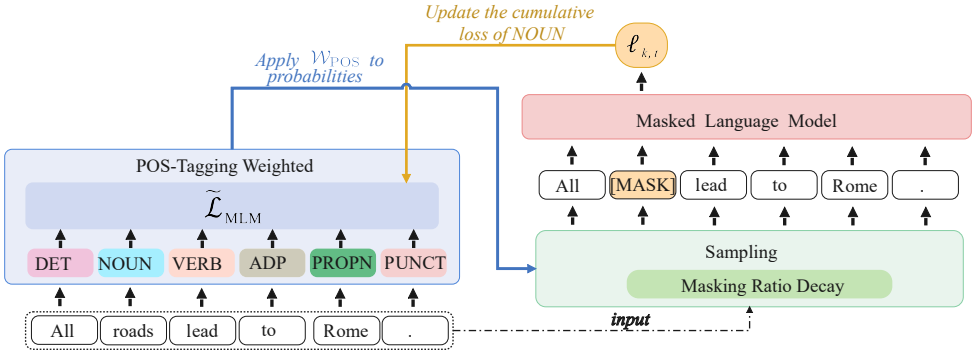


Figure 3: Illustration of our time-variant masking.

algorithm, the degree of acceptance of suboptimal solutions depends on the annealing temperature T according to the Metropolis algorithm (Metropolis et al., 1953). That is, the higher the temperature parameter T is, the larger the solution space allowed to be explored. Thus, the model can easily jump out the local minima if the T is large. As the model converges and the annealing temperature decreases, the intolerance of suboptimal solution rises and a better local optimal solution can be found. In MRD, the magnitude of annealing temperature T can be analogous to the masking ratio $p\%$. A high masking ratio means less information in the input sequence, allowing the model to explore more possibilities on coarse-grained task. On the other hand, a low masking rate allows the model to focus on finding a better solution close to global minima on the fine-grained task.

3.3 POS-Tagging Weighted (PTW) Masking

In this section, on the basis of the discussion of Section 2.3, we present the POS-Tagging Weighted (PTW) Masking, making the models have more chance to train on the difficult words according to the current training state.

Firstly, in the data pre-processing part, we use POS-tagging tools to label the words with corresponding part-of-speech. We then use WordPiece Tokenizer to perform tokenization and align tokens with their part-of-speech tags while ignoring the special tokens [CLS], [SEP], and [PAD].

Before training batch starts, we first apply PTW Masking to corrupt the sequences, while the masking ratio remains unchanged. When the model is trained at t steps, according to Equation 2, we can obtain cumulative loss vector $\tilde{\mathcal{L}}_{\text{MLM}} = \{\tilde{\ell}_1, \tilde{\ell}_2, \ldots, \tilde{\ell}_k, \ldots, \tilde{\ell}_m\}, k \in \mathcal{C}$ for m categories of words, and the cumulative loss vector $\tilde{\mathcal{L}}_{\text{MLM}}$ is converted into the corresponding weight vector $\mathcal{W}_{\text{POS}} = \{w_1, w_2, \ldots, w_k, \ldots, w_m\}, w_k \in (0, 1)$ by the following equation:

$$\mathcal{W}_{\text{POS}} = \text{sigmoid}\left(\frac{\tilde{\mathcal{L}}_{\text{MLM}} - \mathbb{E}(\tilde{\mathcal{L}}_{\text{MLM}})}{\sqrt{Var(\tilde{\mathcal{L}}_{\text{MLM}})} \cdot \mu}\right), \quad (5)$$

where μ is a coefficient to adjust the input for sigmoid function. We apply this weight vector $\mathcal{W}_{\text{POS}}$ to the masking probabilities. Equation 5 is based on Equation 2, where the process of smoothing gives the weights changing relatively stably for masking. We do not use bias correction in Equation 2, which enables the cumulative losses for each kind of words to grow from zero. That is, in the very beginning, the $\mathcal{W}_{\text{POS}}$ is initialized with the same value for each type of words and weights the probabilities for masking equally.

Specifically, the masking probability of each word is weighted by its corresponding part-of-speech, so that words with higher losses are more likely to be masked. We show that non-function words tend to have much higher losses than function words, so the language model learns to model non-function words most of the time, but fewer function words and punctuation, as shown in Figure 1c. In special case, PTW Masking is similar to Named Entities Masking (Sun et al., 2019) if only proper nouns have a weight of 1 and the others are 0.

4 Experiment Setup

4.1 Datasets and Setup

For full details of pre-training and finetuning, please see the Appendix C.

4.1.1 Pre-training

For pre-training, we use the BERT-base and BERT-large model as the representatives of MLMs for

training. Specifically, we train BERT-base for 200k steps from scratch and continue training BERT-large models for 100k steps initialized by pre-trained weights [4]. In practice, we explore masking ratio with MRD and masked content with PTW Masking seperately to avoid mutual influence. For the dataset, we train BERT on English Wikipedia using WordPiece Tokenizer for tokenization.

4.1.2 Finetuning

We finetune our models on GLUE (Wang et al., 2019) and SQuAD v1.1 (Rajpurkar et al., 2016) to evaluate the performance of downstream tasks. To further explore what extra skills models using PTW Masking have learnt, we evaluate BERT-large on CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), which is a widely used NER benchmark, to test the information extraction ability of the models. We train all tasks 5 times respectively and report the average scores. For more detailed introduction of downstream tasks, please see Appendix B.

4.2 Implementation Details

For the implementation of MRD, we train the model using Random-Token Masking with a fixed masking ratio of 15% as the baseline. Then we apply our MRD to Random-Token Masking with an average masking ratio of 15% ($p = 15$) using linear and cosine decay, respectively.

Because there is the parameter T in Equation 3, the number of training batches learned by the models under a certain masking ratio will differ if total training step T is different. As shown in Figure 4, compared to the model with 200k steps, the models with 1M steps are trained for large masking ratio for longer time in early stage. This question will not be raised if we use the fixed masking ratio as we usually do. But in MRD, though both masking ratios decay in relatively the same way, the absolute difference of masking ratio in early stage may affect the performance of downstream task. Thus, we additionally train the models for 1M steps to explore if training on large masking ratio longer time or decaying faster at early stage is more beneficial to pre-training.

For the implementation of PTW Masking, we also use Random-Token Masking with a fixed masking ratio of 15% as baseline. We train the model with PTW Masking with the same fixed ratio of 15% for equal comparisons.

[4] https://huggingface.co/bert-large-uncased

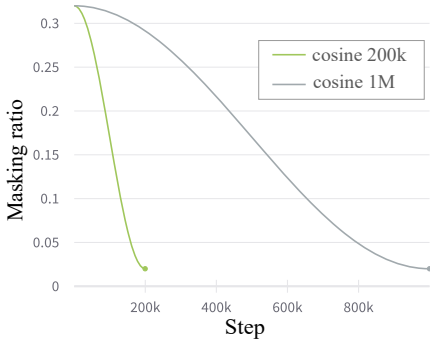


Figure 4: Different total training steps cause an absolute difference in masking ratio in the early stage of pre-training though same MRD strategies are applied.

5 Results

5.1 Results of MRD

The experimental results show that MRD greatly improves downstream task performance and pre-training efficiency (and more discussion on MRD in Appendix A).

5.1.1 Decaying Masking Ratio vs. Fixed Ratio

In Figure 5, we show the SQuAD performance for every 50k checkpoint during pre-training. We observe that the large masking ratio gives a better downstream performance at the start and the decaying mechanism continues to boost the downstream performance, which takes the advantage of high masking ratio and low masking ratio discussed in Section 2.4. The model using cosine decay at 650k steps has obtained a competitive SQuAD v1.1 F1 score to the baseline at 1M steps, thus reducing the training time by 35%.

5.1.2 Influence of Masking Ratio at Early Stage

We further explore the absolute difference (mentioned in Section 4.2) with different training steps in masking ratio using the same MRD strategies. As shown in Table 2, for GLUE tasks, MRD training for 1M steps has more obvious advantages than 200k steps. The model trained with 1M steps performs well above baseline on all subtasks using MRD, with an average increase of 1+ on GLUE, which has a larger increment compared to 200k steps. The comparison shows that models benefit from training for a longer time with a large masking ratio from the start, especially on GLUE. Because the subtasks in GLUE are mainly sequence-level, which focus on global semantics. For a higher masking ratio, the model tries to train

Table 2: Evaluating the performance of models using MRD but trained with two different total training steps. For GLUE, we report STS using Spearman correlation and CoLA using Matthew's correlation, and other tasks using accuracy. And we report SQuAD using F1.

Model	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	GLUE Avg	SQuAD
Masking Ratio Decay										
Trained for 200k steps										
Random-Token 15%	43.8	90.4	85.6	85.9	90.0	**81.5**	88.7	62.2	78.5	87.1
Linear Decay	44.5	**90.5**	**86.0**	85.4	90.0	81.0	89.6	**63.2**	78.8(↑0.3)	87.7(↑0.6)
Cosine Decay	**44.9**	90.3	85.8	**86.1**	**90.2**	81.2	**89.7**	62.6	**78.9**(↑0.4)	**87.7**(↑0.6)
Trained for 1M steps										
Random-Token 15%	54.8	91.5	86.2	86.8	90.3	83.7	90.8	64.7	81.1	90.2
Linear Decay	55.2	92.3	86.8	87.7	90.4	84.4	91.2	**70.4**	82.3(↑1.2)	90.6(↑0.4)
Cosine Decay	**57.2**	**92.3**	**88.2**	**87.8**	**90.5**	**84.7**	**92.3**	69.5	**82.8**(↑1.7)	**90.9**(↑0.7)

Table 3: Evaluating the performance of models using PTW Masking. We report CoNLL using F1.

Model	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	GLUE Avg	SQuAD	CoNLL
PTW Masking											
BERT$_{\text{base}}$ *from scratch*											
Random-Token	43.8	**90.4**	**85.6**	85.9	90.0	81.5	88.7	62.2	78.5	87.1	-
PTW	**45.3**	90.0	85.4	**86.8**	**90.1**	**82.0**	**91.1**	**62.5**	**79.2**(↑0.7)	**88.1**(↑1.0)	-
BERT$_{\text{large}}$ *from continue-training*											
Random-Token	61.8	92.8	86.5	89.3	91.3	86.3	92.4	67.2	83.4	91.3	94.9
PTW	**62.5**	**93.3**	**86.8**	**90.0**	**91.3**	**86.6**	**92.5**	**68.2**	**83.9**(↑0.5)	**91.6**(↑0.3)	**95.4**(↑0.5)

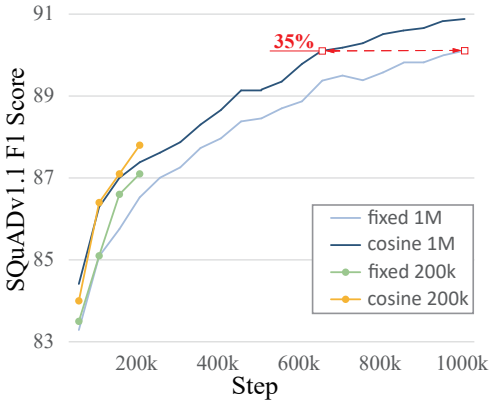


Figure 5: Comparisons between fixed ratio and cosine decay strategy on SQuAD performance during pre-training. We evaluate the saved checkpoints for every 50k steps on SQuAD v1.1 dev set following the same experimental setup.

on a coarse-grained task, inferring global semantics from fewer words, which is more suitable for GLUE. Therefore, in the training of 200k steps, the masking ratio decays too fast, resulting in insufficient training on coarse-grained tasks. In contrast, the model with 1M steps can maintain the training at a high masking ratio for a longer time and thus perform better in the sequence-level tasks of GLUE.

5.1.3 Comparisons Between Different MRD

Compared with linear decay, cosine decay has better downstream task performance in most subtasks in GLUE. The difference between these two is that cosine decay keeps a higher masking ratio in the early stage and decays more quickly, which is consistent with the analysis mentioned above. To move forward, it is necessary to maintain a high masking ratio in the early stage of pre-training. According to Section 3.2's empirical analysis, the model can explore a larger global solution space by using a higher masking ratio in the early training period so as to better converge to the optimal global minima when the masking ratio decreases later.

5.2 Results of PTW Masking

Results in Table 3 show that PTW Masking has significantly improved the performance of various downstream tasks.

5.2.1 What Skills Models Have Learnt if Trained with Mostly Non-function Words with PTW Masking?

Because models trained with PTW Masking pay more attention to important words like non-function words, we can see that models are especially good at information extraction tasks, e.g., extractive QA and NER. The gains

in SQuAD and CoNLL in Table 3 have shown models are sensitive to words with more semantic information, which is consistent with the design of the pre-training goal.

Besides that, PTW Masking also does well in several NLU tasks in GLUE. PTW Masking is a time-variant strategy, which aggregates the advantages of both Random-Token Masking and Named Entities Masking. From the start of pre-training, models can learn from all the words equiprobably like Random-Token Masking. And at the later stage, models memorize more knowledge by masking important words instead of wasting time on predicting meanless words. Thus models can have better NLU ability with memorization of more knowledge under the condition of training with the same number of tokens.

6 Related Work

The pre-processing of the MLM is to replace a subset of the tokens in the input with [MASK] tokens, which has two considerations to optimize: how many tokens to mask (masking ratio) and what tokens to mask (masked content).

• **Masking Ratio**. Masking ratio is a very important hyperparameter that affects the pre-training of MLM, which is relatively seldom studied. In BERT, the masking ratio of 15% is the most commonly used value and is also applied in other MLMs. The generator of ELECTRA (Clark et al., 2020) is a MLM, using 15% for base-sized models and 25% for large-sized models. However, considering the cooperation with the discriminator, it is difficult to judge the effect of 25% on MLM. In a recent study, (Wettig et al., 2022) suggests that a masking ratio of 40% performs better than 15% in downstream tasks of RoBERTa-large (Liu et al., 2019) model. T5 (Raffel et al., 2020) uses an MLM-style pre-training method and also experiments on the influence of different masking ratios. They find that the masking ratio has a limited effect on the model's performance except for 50% and use 15% as the final choice. To our best knowledge, most studies on masking ratio compare the performances of downstream tasks at the end of pre-training (Raffel et al., 2020), but few studies pay attention to the dynamic influence of masking ratio during pre-training, which is very interesting. We record the changes in the performance of downstream tasks under different masking ratios and therefore propose the MRD according to the empirical law we observe. Instead of using a fixed masking ratio, we dynamically decay the ratio and find that the performance of MLM can be greatly improved.

• **Masked Content**. Previous studies have explored strategies for masked content to further improve the Random-Token Masking, though nearly all of them focus on how to select coherent enough masked units. (Devlin et al., 2019) proposes Whole-Word Masking, which forces the model to predict complete words instead of WordPiece tokens. Furthermore, SpanBERT (Joshi et al., 2020), n-gram Masking (Levine et al., 2021; Li and Zhao, 2021) and LIMIT-BERT (Zhou et al., 2020) take into account the continuous mask of multiple word combinations, making model predict tokens using the context with long dependencies. ERNIE (Sun et al., 2019) improves pre-training performance by especially masking named entities. Different from all existing MLM improvements, our PTW Masking lets different types of words correspondingly receive the matched learning intensity, which pioneers a new technical line for the concerned MLM.

7 Conclusion

Masked language model pre-training can be generally defined by two main factors, masking ratio and masked contents. The Random-Token Masking scheme adopted by existing studies treats all words equally and maintains a fixed ratio throughout the entire pre-training, which has been shown suboptimal in our analysis. To better unleash the strength of MLM, we explore two kinds of time-variant masking strategies, namely, Masking Ratio Decay (MRD) and POS-Tagging Weighted (PTW) Masking. Experimental results verify our hypothesis that MLM benefits from time-variant setting both in masking ratio and masked content according to dynamic training states. Our further analysis show that these two time-variant masking schedules greatly improve pre-training efficiency and the performance of downstream tasks.

Limitations

One limitation of this work is that both PTW Masking and MRD are conducted only on BERT due to limited resources, and MLMs with other structures may have different reactions to the time-variant masking with different contents and ratios. Another limitation is that although we propose MRD for the first time, the strategy of time-variant

masking ratio is hard to design like learning rate decay. In fact, other decay methods and choices of starting and ending point are various, where better strategies may exist and further work can be done.

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 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv preprint*, abs/2002.08909.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*, 220(4598):671–680.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. PMI-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*.

Yian Li and Hai Zhao. 2021. Pre-training universal language representation. 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 5122–5133, Online. Association for Computational Linguistics.

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

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper.pdf*.

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. *Journal of Machine Learning Research*, 21:1–67.

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.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *ArXiv preprint*, abs/1904.09223.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. 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.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in

masked language modeling? *arXiv preprint arXiv:2202.08005*.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2021. Dict-bert: Enhancing language model pre-training with dictionary. *arXiv preprint arXiv:2110.06490*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

A Additional Investigation on MRD

A.1 Magnitude of Masking Ratio in MRD

When using MRD, we explore the influence of the much higher masking ratios, which affect the downstream performance of the model. Previous studies (Raffel et al., 2020; Wettig et al., 2022) have shown that a much higher fixed masking ratio ($\gg 40\%$) will cause significant degradation in the model performance because the model can only infer from a small amount of known information resulting in quickly converging to local minima. In MRD, we show that the design of the decaying mechanism can mitigate the impact of the high masking ratio. For the BERT-base model, starting from a high ratio (30%) and a much higher ratio (55%), both can outperform the baseline with a similar margin. We show that higher masking ratios in early pre-training stage help downstream performance, and MRD prevents the high masking ratios from destroying pre-training in later stage.

A.2 MRD Interacts with Learning Rate

Moreover, we show a subtle relationship between MRD and learning rate decay. When the masking ratio is low, using a relatively high learning rate will cause a huge decline in model performance. Therefore, in MRD, the masking ratio and the learning rate both adopt the same type of decay strategy except that the learning rate has an additional warmup stage. For example, cosine masking ratio decay uses cosine learning rate decay.

A.3 Other Simple Schedules in MRD

Based on the same experiment setup, we train the models with other simple schedules (shown in Figure 6) for 200k steps using the linear learning rate decay and finetune them on the SQuAD v1.1. The results on SQuAD v1.1 dev set are presented in Table 4. We find that cosine is the best compared with those alternatives.

Model	**SQuAD v1.1** EM	F1
Masking Ratio Decay		
Random 15%	79.4	87.1
$-ax^2(a>0)$ Decay	79.8	87.6
$ax^2(a>0)$ Decay	79.3	87.2
Ascending	79.6	87.4
Ascending then Decaying	79.4	86.9
Linear Decay	79.8	87.7
Cosine Decay	**79.9**	**87.7**

Table 4: Results of other simple schedules on the SQuAD v1.1 dev set.

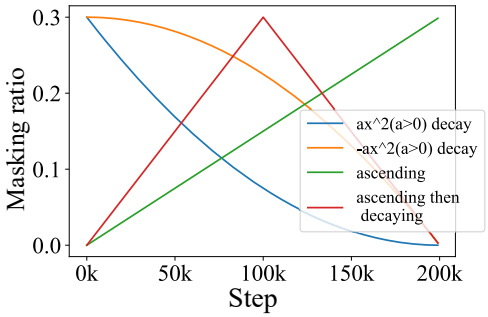


Figure 6: Other simple schedules of adjusting the masking ratios.

B Finetuning Benchmarks

GLUE The General Language Understanding Evaluation (GLUE) benchmark is a collection of 9 various tasks for sequence-level classification for evaluating natural language understanding systems.

SQuAD The Stanford Question Answering Dataset (SQuAD) is a commonly used benchmark for question answering. The task is to predict the text span of an answer from a given passage-question pair.

CoNLL The CoNLL-2003 concerns language-independent named entity recognition. It concen-

trates on four types of named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups.

C Pre-training and Finetuning Details

C.1 Pre-training Details

We only use the MLM task as the training objective and discard the Next Sentence Prediction task, as it has been shown to be redundant in previous studies (Liu et al., 2019; Joshi et al., 2020).

Hyperparameter	Base	Large
Learning Rate	2e-4	5e-5
Warmup Steps	10000	10000
Weight Decay	0.01	0.01
Batch size	256	256
Sequence Length	512	512
Gradient Clipping	1.0	1.0

Table 5: Pre-training hyperparameters for BERT models.

C.2 Finetuning Details

Following the common finetuning practice, we do not use any additional training strategies. We train all tasks 5 times respectively and report the average scores. For GLUE, We use 8 widely used tasks in GLUE.[5]

Hyperparameter	Base	Large
GLUE		
Learning Rate	{5e-5, 1e-4}	{1e-5, 2e-5}
Batch Size	32	32
Weight Decay	0	0
Training Epochs*	3	3
SQuAD v1.1		
Learning Rate	{5e-5, 1e-4}	{2e-5, 5e-5}
Batch Size	128	32
Weight Decay	0	0
Training Epochs	3	3
CoNLL-2003		
Learning Rate	-	{2e-5, 5e-5}
Batch Size	-	32
Weight Decay	-	0
Training Epochs	-	3

Table 6: Finetuning hyperparameters for BERT models. *We finetune our models in RTE and STS-B for 10 epochs and other subtasks for 3 epochs.

[5]For a fair comparison, we exclude the WNLI following the previous work (Devlin et al., 2019).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

4

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.