Pre-Training With Whole Word Masking for Chinese BERT

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang

***Abstract*—Bidirectional Encoder Representations from Transformers (BERT) has shown marvelous improvements across various NLP tasks, and its consecutive variants have been proposed to further improve the performance of the pre-trained language models. In this paper, we aim to first introduce the *whole word masking* (wwm) strategy for Chinese BERT, along with a series of Chinese pre-trained language models. Then we also propose a simple but effective model called MacBERT, which improves upon RoBERTa in several ways. Especially, we propose a new masking strategy called *MLM as correction* (Mac). To demonstrate the effectiveness of these models, we create a series of Chinese pre-trained language models as our baselines, including BERT, RoBERTa, ELECTRA, RBT, etc. We carried out extensive experiments on ten Chinese NLP tasks to evaluate the created Chinese pre-trained language models as well as the proposed MacBERT. Experimental results show that MacBERT could achieve state-of-the-art performances on many NLP tasks, and we also ablate details with several findings that may help future research. We open-source our pre-trained language models for further facilitating our research community.[1]**

***Index Terms*—Pre-trained language model, representation learning, natural language processing.**

I. Introduction

BERT [2] has become enormously popular and has proven to be effective in recent natural language processing studies, which utilizes large-scale unlabeled training data and generates enriched contextual representations. As we traverse several popular machine reading comprehension benchmarks, such as SQuAD [3], CoQA [4], QuAC [5], NaturalQuestions [6], RACE [7], we can see that most of the top-performing models are based on BERT and its variants [8]–[10], demonstrating that the pre-trained language models have become new fundamental components in natural language processing field.

Starting from BERT, the community members have made great and rapid progress on optimizing the pre-trained language models, such as ERNIE [11], XLNet [12], RoBERTa [13], SpanBERT [14], ALBERT [15], ELECTRA [16], etc. However, training Transformer-based [17] pre-trained language models are not as easy as we used to train word embeddings or other traditional neural networks for learning representations. Typically, training a powerful BERT-large model with a 24-layer Transformer and 330 million parameters, to convergence needs high-memory computing devices, such as TPU or TPU Pod, which are very expensive. On the other hand, though various pre-trained language models have been released, most of them are based on English, and there are few efforts on building powerful pre-trained language models in other languages.

To minimize the repetitive work and build baselines for future studies, in this paper, we aim to build Chinese pre-trained language model series and release them to the public for facilitating the research community, as Chinese and English are among the most spoken languages in the world. We revisit the existing popular pre-trained language models and adjust them to the Chinese language to see whether these models could generalize and perform well in a language other than English. Besides, we also propose a new pre-trained language model called MacBERT, which replaces the original MLM task into **M**LM **a**s **c**orrection (Mac) task. MacBERT mainly aims to mitigate the discrepancy of the pre-training and fine-tuning stage in original BERT. Extensive experiments are conducted on ten popular Chinese NLP datasets, ranging from sentence-level to document-level tasks, such as machine reading comprehension, text classification, etc. The results show that the proposed MacBERT could give significant gains in most of the tasks against other pre-trained language models, and detailed ablations are given to better examine the composition of the improvements. The contributions of this paper are listed as follows.

- To further accelerate future research on Chinese NLP, we create and release the Chinese pre-trained language model series to our community. Extensive empirical studies are carried out to revisit the performance of these pre-trained language models on various tasks with careful analyses.
- We propose a new pre-trained language model called MacBERT that mitigates the gap between the pre-training and fine-tuning stage by masking the word with its similar word, which has proven to be effective on various downstream tasks.
- We also create a series of small models, called RBT, to demonstrate how small models perform compared to regular pre-trained language models, which could help utilize them in real-life applications.

Manuscript received March 7, 2021; revised July 27, 2021 and October 18, 2021; accepted October 25, 2021. Date of publication November 2, 2021; date of current version December 3, 2021. The work of Yiming Cui was supported in part by the Google TPU Research Cloud program for Cloud TPU access. This work was supported by the National Key R&D Program of China under Grant 2020AAA0106501 and the National Natural Science Foundation of China under Grants 61976072 and 61772153. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhijian Ou. *(Corresponding author: Ting Liu.)*

Yiming Cui, Wanxiang Che, Ting Liu, and Bing Qin are with the Harbin Institute of Technology, Harbin 150001, China (e-mail: ymcui@ir.hit.edu.cn; car@ir.hit.edu.cn; tliu@hit.edu.cn; qinb@ir.hit.edu.cn).

Ziqing Yang is with the State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Beijing 100010, China (e-mail: zqyang5@iflytek.com).

Digital Object Identifier 10.1109/TASLP.2021.3124365

[1]An extended version of [1]. The resources are available through https://github.com/ymcui/Chinese-BERT-wwm

TABLE I
COMPARISONS OF THE PRE-TRAINED LANGUAGE MODELS. (AE: AUTO-ENCODING, AR: AUTO-REGRESSIVE, T: TOKEN, S: SEGMENT, P: POSITION, E: ENTITY, PH: PHRASE, WWM: WHOLE WORD MASKING, NM: N-GRAM MASKING, NSP: NEXT SENTENCE PREDICTION, SOP: SENTENCE ORDER PREDICTION, MLM: MASKED LM, PLM: PERMUTATION LM, GEN-DIS: GENERATOR-DISCRIMINATOR, MAC: MLM AS CORRECTION)

	BERT	**ERNIE**	**XLNet**	**RoBERTa**	**ALBERT**	**ELECTRA**	**MacBERT**
Type	AE	AE	AR	AE	AE	AE	AE
Embeddings	T/S/P	T/S/P	T/S/P	T/S/P	T/S/P	T/S/P	T/S/P
Masking	T	T/E/Ph	-	T	T	T	WWM/NM
LM Task	MLM	MLM	PLM	MLM	MLM	Gen-Dis	Mac
Paired Task	NSP	NSP	-	-	SOP	-	SOP

II. RELATED WORK

In this section, we revisit the techniques of the representative pre-trained language models in the recent natural language processing field. The overall comparisons of these models, as well as the proposed MacBERT, are depicted in Table I. We elaborate on their key components in the following subsections.

A. BERT

BERT (Bidirectional Encoder Representations from Transformers) [2] has demonstrated its effectiveness in a wide range of natural language processing tasks. BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all Transformer layers. Primarily, BERT consists of two pre-training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

- *MLM:* Randomly masks some of the tokens from the input, and the objective is to predict the original word based only on its context.
- *NSP:* To predict whether sentence *B* is the next sentence of sentence *A*.

Later, they further propose a technique called whole word masking (wwm) for optimizing the original masking in the MLM task. In this setting, instead of randomly selecting WordPiece [18] tokens to mask, we always mask all of the tokens corresponding to a whole word at once. This explicitly forces the model to recover the whole word in the MLM pre-training task instead of just recovering WordPiece tokens [1], which is much more challenging. As the whole word masking only affects the masking strategy of the pre-training process, it would not bring additional burdens on downstream tasks. Moreover, as training pre-trained language models are computationally expensive, they also release all the pre-trained models as well as the source codes, which significantly stimulates the community to have great interests in the research of pre-trained language models.

B. ERNIE

ERNIE (Enhanced Representation through kNowledge IntEgration) [11] is designed to optimize the masking process of BERT, which includes entity-level masking and phrase-level masking. Different from selecting random words in the input, entity-level masking masks the named entities, which are often formed by several words. Phrase-level masking is to mask consecutive words, which is similar to the N-gram masking strategy [2], [14], [19].[2]

C. XLNet

[12] argues that the existing pre-trained language models that are based on auto-encoding, such as BERT, which suffer from the discrepancy of the pre-training and fine-tuning stage because the masking token `[MASK]` never appears in the fine-tuning stage. To alleviate this problem, XLNet is proposed, which is based on Transformer-XL [8]. XLNet mainly modifies in two ways. The first is to maximize the expected likelihood over all permutations of the factorization order of the input, where they call the Permutation Language Model. To achieve this goal, they propose a novel two-stream self-attention mechanism. Another one is to change the auto-encoding language model into an auto-regressive one, which is similar to the traditional statistical language models.

D. RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) [13] aims to adopt original BERT architecture but make much more precise modifications to fully release the power of BERT, which is underestimated in [2]. They carry out careful comparisons of various components in BERT, including the masking strategies, input format, training steps, etc. After thorough evaluations, they come up with several useful conclusions to make BERT more powerful, mainly including 1) training longer with bigger batches and longer sequences over more data; 2) removing the next sentence prediction task and using dynamic masking in MLM task.

E. ALBERT

ALBERT (A Lite BERT) [15] primarily tackles the problems of higher memory consumption and slow training speed of BERT. ALBERT introduces two techniques for parameter reduction. The first one is the factorized embedding parameterization, which decomposes the embedding matrix into two small matrices. The second one is the cross-layer parameter sharing that the Transformer weights are shared across each layer of ALBERT, which significantly reduces the overall parameters. Besides, they also propose the sentence-order prediction (SOP)

[2]Though N-gram masking was not included in [2], according to their model name in SQuAD leaderboard, we often admit their credit towards this method.

task to replace the traditional NSP pre-training task and yield better performances.

F. ELECTRA

ELECTRA (Efficiently Learning an Encoder that Classifiers Token Replacements Accurately) [16] employs a new generator-discriminator framework that is similar to generative adversarial net (GAN) [20]. The generator is typically a small MLM that learns to predict the original words of the masked tokens. The discriminator is trained to discriminate whether the input token is replaced by the generator, which they call Replaced Token Detection (RTD). Note that, to achieve efficient training, the discriminator is only required to predict a binary label to indicate "replacement," unlike the way of MLM that should predict the exact masked word. After the pre-training stage, we discard the generator and only use the discriminator for fine-tuning downstream tasks.

III. Chinese Pre-Trained Language Models

While BERT and its variants have achieved significant improvements in various English tasks, we wonder if these models and techniques could generalize well in other languages. In this section, we illustrate how the existing pre-trained language models are adapted for the Chinese language. We adopt BERT, RoBERTa, and ELECTRA as well as their variants to create Chinese pre-trained model series, and their effectiveness is shown in Section VI. Note that, as these models are all originated from BERT or ELECTRA without changing the nature of the input, no modification should be made to adapt to these models in the fine-tuning stage, which is very flexible for replacing one another.

A. BERT-wwm & RoBERTa-wwm

In the original BERT, a WordPiece tokenizer [18] is used to split the text into WordPiece tokens, where some words are split into several small fragments. The whole word masking (wwm) mitigates the drawback of masking only a part of the whole word, which is easier for the model to predict. In Chinese condition, WordPiece tokenizer no longer splits the word into small fragments, as Chinese characters are not formed by alphabet-like symbols. We use the traditional Chinese Word Segmentation (CWS) tool to split the text into several words. In this way, we could adopt the whole word masking in Chinese to mask the word instead of individual Chinese characters. For implementation, we strictly follow the original whole word masking codes and do not change other components, such as the percentage of word masking, etc. We use LTP [21] for Chinese word segmentation to identify the word boundaries. Note that the whole word masking only affects the selection of the masking tokens in the pre-training stage. We still use WordPiece tokenizer to split the text, which is identical to the original BERT.

Similarly, whole word masking can also be applied on RoBERTa, where the NSP task is not adopted. However, we still use a paired input for pre-training, which could be beneficial to the sentence pair classification and reading comprehension tasks. An example of the whole word masking is depicted in Table II.

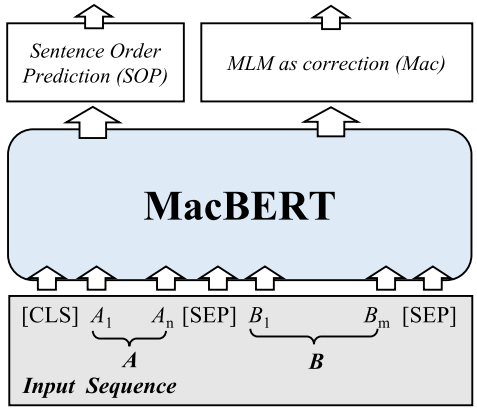


Fig. 1. Neural architecture of MacBERT.

B. ELECTRA

Besides BERT and RoBERTa series, we also explore the ELECTRA model, which adopts a new pre-training framework that consists of a generator and discriminator. We strictly follow the original implementation as in [16].

C. RBT Series

Though the aforementioned pre-trained language models are powerful, they are computationally ineffective and hard to adopt in real-life applications. To make pre-trained models more accessible by community researchers, besides the regular pre-trained language models, we also pre-train several small models, where we call RBT. Specifically, we use exactly the same training strategy as in training RoBERTa, but we use fewer Transformer layers. We train 3-layer, 4-layer, 6-layer RoBERTa-base, denoted as RBT3, RBT4, and RBT6, respectively. We also train a 3-layer RoBERTa-large, denoted as RBTL3, which has a similar parameter size as RBT6. This is designed to compare a wider and shorter model (RBTL3) with a thinner and taller model (RBT6) under a comparable parameter size, which could be useful in the design of future pre-trained language models.

IV. MacBERT

In the previous section, we propose a series of Chinese pre-trained language models. In this section, we make the best use of them and propose a novel model called **MacBERT** (**M**LM **a**s **c**orrection **BERT**). MacBERT shares the similar types of pre-training tasks as BERT with several modifications. MacBERT consists of two pre-training tasks: MLM as correction, and sentence order prediction. The overall architecture of MacBERT is depicted in Fig. 1.

TABLE II
EXAMPLES OF DIFFERENT MASKING STRATEGIES. WE ALSO INCLUDE AN ENGLISH EXAMPLE FOR CLARITY. MASKED TOKENS ARE IN BOLDFACE

	Chinese	English
Original Sentence	使用语言模型来预测下一个词的概率。	we use a language model to predict the probability of the next word.
+ CWS	语言 模型 来 预测 下 一个 词 的 概率 。	-
+ BERT Tokenizer	语 言 模 型 来 预 测 下 一 个 词 的 概 率 。	we use a language **model** to **pre ##di ##ct** the **pro ##ba ##bility** of the next word .
Original Masking	语 言 **[M]** 型 来 **[M]** 测 下 一 个 词 的 概 率 。	we use a language **[M]** to **[M] ##di ##ct** the **pro [M] ##bility** of the next word .
+ WWM	语 言 **[M] [M]** 来 **[M] [M]** 下 一 个 词 的 概 率 。	we use a language **[M]** to **[M] [M] [M]** the **[M] [M] [M]** of the next word .
++ N-gram Masking	**[M] [M] [M] [M]** 来 **[M] [M]** 下 一 个 词 的 概 率 。	we use a **[M] [M]** to **[M] [M] [M]** the **[M] [M] [M] [M] [M]** next word .
+++ Mac Masking	语 法 建 模 来 预 见 下 一 个 词 的 几 率 。	we use a **text system** to **ca ##lc ##ulate** the **po ##si ##bility** of the next word .

A. MLM as Correction

Masked Language Model (MLM) is the most important pre-training task in BERT and its variants, which models bidirectional contextual inference ability. However, as shown in the previous section, the MLM suffers from the 'pre-training and fine-tuning' discrepancy, where the artificial tokens in the pre-training stage, such as `[MASK]`, never appear in the real downstream fine-tuning tasks.

To address this issue, we propose a novel pre-training task called MLM as correction (Mac). In this pre-training task, we do not adopt any pre-defined tokens for masking purposes. Instead, we transform the original MLM as a text correction task, where the model should correct the wrong word into the correct one, which is much more natural than MLM. Specifically, in the Mac task, we perform the following modifications on the original MLM.

- We use the whole word masking as well as N-gram masking strategies to select candidate tokens for masking, with a percentage of 40%, 30%, 20%, 10% for word-level unigram to 4-gram. We also notice that a recent work PMI-masking [22] is proposed, which optimizes the masking strategy. In this paper, we resort to vanilla N-gram masking and will try PMI-masking in the future.
- Instead of masking with `[MASK]` token, which never appears in the fine-tuning stage, we propose to use similar words for the masking purpose. A similar word is obtained by using *Synonyms* toolkit[3], which is based on word2vec [23] similarity calculations. If an N-gram is selected to mask, we find similar words individually. In rare cases, when there is no similar word, we degrade to use random word replacement. Such replacements are restricted to no more than 10% of all tokens to be masked.
- Following previous works, we use a percentage of 15% input words for masking, where 80% tokens are replaced with similar words, 10% tokens are replaced with random words, and keep with original words for the rest of 10%.

B. Sentence Order Prediction

The original next sentence prediction (NSP) task in BERT is considered to be too easy for the model and proved to be not that effective [13], [15]. In this paper, we adopt the sentence order prediction (SOP) task as introduced by ALBERT [15], which is shown to be much more effective than NSP. The positive samples are created by using two consecutive texts, while the negative ones are created by switching the original order of them. We ablate these modifications in Section VII-A to better demonstrate the contributions of each component.

[3][Online]. Available: https://github.com/huyingxi/Synonyms

C. Neural Architecture

Formally, given a pair of sequences $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$, we first construct the input sequence X by concatenating two sequences. Then, MacBERT converts X into a contextualized representation $\boldsymbol{H}^{(L)} \in \mathbb{R}^{N \times d}$ through an embedding layer (which consists of word embedding, positional embedding, and token type embedding), and a consecutive L-layer transformer, where N is the maximum sequence length, and d is the dimension of hidden layers.

$$X = [\texttt{CLS}]\, A_1 \ldots A_n\, [\texttt{SEP}]\, B_1 \ldots B_m\, [\texttt{SEP}] \tag{1}$$

$$\boldsymbol{H}^{(0)} = \mathbf{Embedding}(X) \tag{2}$$

$$\boldsymbol{H}^{(i)} = \mathbf{Transformer}(\boldsymbol{H}^{(i-1)}),\ i \in \{1, \ldots, L\} \tag{3}$$

As we only need to predict the positions that are replaced by the Mac task, after getting the contextual representation $\boldsymbol{H}^L$, we collect a subset with respect to the replaced positions, forming the replaced representation $\boldsymbol{H}^{\mathrm{m}} \in \mathbb{R}^{k \times d}$, where k is the number of the replaced tokens. According to the definition of Mac task, $k = \lfloor N \times 15\% \rfloor$.

Then we project $\boldsymbol{H}^{\mathrm{m}}$ into the vocabulary space to predict the probability distributions $\boldsymbol{p}$ over the whole vocabulary $\mathbb{V}$. Following original BERT implementation, we also use word embedding matrix $\boldsymbol{W}^{\mathrm{e}} \in \mathbb{R}^{|\mathbb{V}| \times d}$ to perform the projection, as the embedding and hidden size are identical.

$$\boldsymbol{p}_i = \boldsymbol{H}_i^{\mathrm{m}} \boldsymbol{W}^{\mathrm{e}^\top} + \boldsymbol{b} \tag{4}$$

Then we use the standard cross-entropy loss to optimize the pre-training task.

$$\mathcal{L} = -\frac{1}{M} \sum_{i=1}^{M} \boldsymbol{y}_i \log \boldsymbol{p}_i \tag{5}$$

For the SOP task, we directly use the contextual representation of the `[CLS]` token, which is the first component of $\boldsymbol{H}$, and project it into the label prediction layer.

$$\boldsymbol{p} = \mathbf{softmax}(\boldsymbol{h}_0 \boldsymbol{W}^{\mathrm{s}} + \boldsymbol{b}^{\mathrm{s}}) \tag{6}$$

where the $\boldsymbol{W}^{\mathrm{s}} \in \mathbb{R}^{d \times 2}$ and $\boldsymbol{b}^{\mathrm{s}} \in \mathbb{R}^2$ are the weight matrix and bias. Then we also use the cross-entropy loss to optimize the

TABLE III
TRAINING DETAILS OF CHINESE PRE-TRAINED LANGUAGE MODELS

	BERT	BERT-wwm	RoBERTa-wwm	RBT	ELECTRA	MacBERT
Word #	0.4B	5.4B	5.4B	5.4B	5.4B	5.4B
Vocab #	21,128	21,128	21,128	21,128	21,128	21,128
Hidden Activation	GeLU	GeLU	GeLU	GeLU	GeLU	GeLU
Optimizer	AdamW	LAMB	AdamW	AdamW	AdamW	LAMB
Training Steps (base/large)	?	2M	1M / 2M	1M	1M / 2M	1M / 2M
Initial Checkpoint (base)	random	BERT	BERT	RoBERTa	random	BERT

pre-training task (similar to (5)). Finally, the overall training loss is the combination of the Mac and SOP task.

$$\mathcal{L} = \mathcal{L}_{mac} + \mathcal{L}_{sop} \tag{7}$$

V. EXPERIMENTAL SETUPS

A. Data Processing

We use Wikipedia dump[4] (as of March 25, 2019), and pre-process with `WikiExtractor.py` as suggested by [2], resulting in 1,307 extracted files. We use both Simplified and Traditional Chinese in this dump and do not convert the Traditional Chinese portion into Simplified one. We demonstrate the effectiveness of the Traditional Chinese task in Section VI-A. After cleaning the raw text, such as removing `html` tags and separating the document, we obtain about 0.4B words. As Chinese Wikipedia data is relatively small, besides Chinese Wikipedia, we also use extended training data for training these pre-trained language models (mark with `ext` in the model name). The in-house collected extended data contains encyclopedia, news, and question answering web, which has 5.4B words and is over ten times bigger than the Chinese Wikipedia. Note that we always use extended data for MacBERT and omit the `ext` mark. In order to identify the boundary of Chinese words for whole word masking, we use LTP [21] for Chinese word segmentation. We use official `create_pretraining_data.py` provided by [2] to convert the raw input text to the pre-training examples.

B. Setups for Pre-Trained Language Models

To better acquire the knowledge from the existing pre-trained language model, we did NOT train our base-level model from scratch but the official Chinese BERT-base, inheriting its vocabulary and weight. However, for the large-level model, we have to train from scratch but still use the same vocabulary provided by the base-level model. The base-level model is a 12-layer transformer with a hidden dimension of 768, while the large-level model is a 24-layer transformer with a hidden dimension of 1024.

For training BERT series, we adopt the scheme of training on a maximum sequence length of 128 tokens then on 512, suggested by [2]. However, we empirically found that this results in insufficient adaptation for the long-sequence tasks, such as reading comprehension. In this context, for models other than BERT, we directly use a maximum length of 512 throughout the pre-training process, which is adopted in [13].

TABLE IV
DATA STATISTICS AND HYPER-PARAMETER SETTINGS FOR DIFFERENT FINE-TUNING TASKS

Dataset	MaxLen	Epoch	LR	Train	Dev	Test
CMRC 2018	512	2	3e-5	10K	3.2K	4.9K
DRCD	512	2	3e-5	27K	3.5K	3.5K
CJRC	512	2	4e-5	10K	3.2K	3.2K
ChnSentiCorp	256	3	2e-5	9.6K	1.2K	1.2K
THUCNews	512	3	2e-5	50K	5K	10K
TNEWS	128	3	2e-5	53.3K	10K	10K
XNLI	128	2	3e-5	392K	2.5K	5K
LCQMC	128	3	2e-5	240K	8.8K	12.5K
BQ Corpus	128	3	3e-5	100K	10K	10K
OCNLI	128	3	2e-5	56K	3K	3K

For smaller batch sizes, we adopt the original ADAM [24] with weight decay optimizer in BERT for optimization, and use LAMB optimizer [25] for better scalability in larger batch size. The pre-training was either done on a single Google Cloud TPU[5] v3-8 (equals to a single TPU) or TPU Pod v3-32 (equals to 4 TPUs), depending on the magnitude of the model. Specifically, for MacBERT-large, we trained for 2M steps with a batch size of 512 and an initial learning rate of 1e-4.

The training details are shown in Table III. For clarity, we do not list '`ext`' models, where the other parameters are the same as the one that is not trained on extended data.

C. Setups for Fine-Tuning Tasks

To thoroughly test these pre-trained language models, we carry out extensive experiments on various natural language processing tasks, covering a wide spectrum of text length, i.e., from sentence-level to document-level. Task details are shown in Table IV. Specifically, we choose the following ten popular Chinese datasets.

- *Machine Reading Comprehension (MRC):* CMRC 2018 [26], DRCD [27], CJRC [28].
- *Single Sentence Classification (SSC):* ChnSentiCorp [29], THUCNews [30], TNEWS [31].
- *Sentence Pair Classification (SPC):* XNLI [32], LCQMC [33], BQ Corpus [34], OCNLI [35].

In order to make a fair comparison, for each dataset, we keep the same hyper-parameters (such as maximum length, warm-up steps, etc.) and only tune the initial learning rate from 1e-5 to 5e-5 for each task. Note that the initial learning rates are tuned on the original Chinese BERT, and it would be possible

[4][Online]. Available: https://dumps.wikimedia.org/zhwiki/latest/

[5][Online]. Available: https://cloud.google.com/tpu/

TABLE V
RESULTS ON CMRC 2018 (SIMPLIFIED CHINESE) AND DRCD. THE AVERAGE SCORES OF 10 INDEPENDENT RUNS ARE DEPICTED IN BRACKETS. OVERALL BEST PERFORMANCES ARE DEPICTED IN BOLDFACE (BASE-LEVEL AND LARGE-LEVEL ARE MARKED INDIVIDUALLY)

	CMRC 2018						DRCD			
	Dev		Test		Challenge		Dev		Test	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BERT	65.5 (64.4)	84.5 (84.0)	70.0 (68.7)	87.0 (86.3)	18.6 (17.0)	43.3 (41.3)	83.1 (82.7)	89.9 (89.6)	82.2 (81.6)	89.2 (88.8)
BERT-wwm	66.3 (65.0)	85.6 (84.7)	70.5 (69.1)	87.4 (86.7)	21.0 (19.3)	47.0 (43.9)	84.3 (83.4)	90.5 (90.2)	82.8 (81.8)	89.7 (89.0)
BERT-wwm-ext	67.1 (65.6)	85.7 (85.0)	71.4 (70.0)	87.7 (87.0)	24.0 (20.0)	47.3 (44.6)	85.0 (84.5)	91.2 (90.9)	83.6 (83.0)	90.4 (89.9)
RoBERTa-wwm-ext	67.4 (66.5)	87.2 (86.5)	72.6 (71.4)	89.4 (88.8)	26.2 (24.6)	51.0 (49.1)	86.6 (85.9)	92.5 (92.2)	85.6 (85.2)	92.0 (91.7)
ELECTRA-base	68.4 **(68.0)**	84.8 (84.6)	73.1 **(72.7)**	87.1 (86.9)	22.6 (21.7)	45.0 (43.8)	87.5 (87.0)	92.5 (92.3)	86.9 (86.6)	91.8 (91.7)
MacBERT-base	**68.5** (67.3)	**87.9 (87.1)**	**73.2** (72.4)	**89.5 (89.2)**	**30.2 (26.4)**	**54.0 (52.2)**	**89.4 (89.2)**	**94.3 (94.1)**	**89.5 (88.7)**	**93.8 (93.5)**
ELECTRA-large	69.1 (68.2)	85.2 (84.5)	73.9 (72.8)	87.1 (86.6)	23.0 (21.6)	44.2 (43.2)	88.8 (88.7)	93.3 (93.2)	88.8 (88.2)	93.6 (93.2)
RoBERTa-wwm-ext-large	68.5 (67.6)	88.4 (87.9)	74.2 (72.4)	90.6 (90.0)	31.5 **(30.1)**	60.1 (57.5)	89.6 (89.1)	94.8 (94.4)	89.6 (88.9)	94.5 (94.1)
MacBERT-large	**70.7 (68.6)**	**88.9 (88.2)**	**74.8 (73.2)**	**90.7 (90.1)**	**31.9** (29.6)	**60.2 (57.6)**	**91.2 (90.8)**	**95.6 (95.3)**	**91.7 (90.9)**	**95.6 (95.3)**

to achieve another gain by tuning the learning rate individually. We run the same experiment ten times to ensure the reliability of the results. The best initial learning rate is determined by selecting the best average development set performance. We report the maximum and average scores to both evaluate the peak and average performance. Except for TNEWS and OCNLI, where the test sets are not publicly available, we report both development and test set results.

For all models except for ELECTRA, we use the same initial learning rate setting for each task, as depicted in Table IV. For ELECTRA models, we use a universal initial learning rate of 1e-4 for base-level models and 5e-5 for large-level models as suggested in [16].

As the pre-training data are quite different among various existing Chinese pre-trained language models, such as ERNIE [11], ERNIE 2.0 [36], NEZHA [37], we only compare BERT [2], BERT-wwm, BERT-wwm-ext, RoBERTa-wwm-ext, RoBERTa-wwm-ext-large, ELECTRA, along with our MacBERT to ensure relatively fair comparisons among different models, where all models are trained by ourselves except for the original Chinese BERT [2]. We carried out experiments under TensorFlow framework [38] with slight modifications to the fine-tuning scripts[6] provided by [2] to better adapt to Chinese tasks.

VI. RESULTS

A. Machine Reading Comprehension

Machine Reading Comprehension (MRC) is a representative document-level modeling task that requires to answer the questions based on the given passages. We mainly test these models on three datasets: CMRC 2018, DRCD, and CJRC.

- *CMRC 2018:* A span-extraction machine reading comprehension dataset, which is similar to SQuAD [39] that extracts a passage span for the given question.
- *DRCD:* This is also a span-extraction MRC dataset but in Traditional Chinese.
- *CJRC:* Similar to CoQA [39], which has yes/no questions, no-answer questions, and span-extraction questions. The data is collected from Chinese law judgment documents. Note that we only use `small-train-data.json` for training.

[6][Online]. Available: https://github.com/google-research/bert

TABLE VI
RESULTS ON CJRC

CJRC	Dev		Test	
	EM	F1	EM	F1
BERT	54.6 (54.0)	75.4 (74.5)	55.1 (54.1)	75.2 (74.3)
BERT-wwm	54.7 (54.0)	75.2 (74.8)	55.1 (54.1)	75.4 (74.4)
BERT-wwm-ext	55.6 (54.8)	76.0 (75.3)	55.6 (54.9)	75.8 (75.0)
RoBERTa-wwm-ext	58.7 (57.6)	79.1 (78.3)	59.0 (57.8)	79.0 (78.0)
ELECTRA-base	59.0 (58.1)	79.4 (78.5)	59.3 (58.2)	79.4 (78.3)
MacBERT-base	**60.4 (59.5)**	**80.3 (79.2)**	**60.3 (59.3)**	**79.8 (79.0)**
ELECTRA-large	61.9 (60.8)	82.1 (81.2)	62.3 (61.2)	82.0 (80.7)
RoBERTa-wwm-ext-L	62.1 (61.1)	**82.4 (81.6)**	62.4 (61.4)	82.2 (81.0)
MacBERT-large	**62.4 (61.3)**	82.3 (81.4)	**62.9 (61.6)**	**82.5 (81.1)**

TABLE VII
RESULTS ON SINGLE SENTENCE CLASSIFICATION TASKS: CHNSENTICORP, THUCNEWS AND TNEWS. 'R' STANDS FOR ROBERTA, 'E' STANDS FOR ELECTRA, 'M' STANDS FOR 'MACBERT'

	ChnSentiCorp		THUCNews		TNEWS
	Dev	Test	Dev	Test	Dev
BERT	94.7 (94.3)	95.0 (94.7)	97.7 (97.4)	**97.8 (97.6)**	56.3 (56.1)
BERT-w	95.1 (94.5)	95.4 **(95.0)**	98.0 (97.6)	**97.8 (97.6)**	56.5 (56.3)
BERT-w-e	**95.4** (94.6)	95.3 (94.8)	97.7 (97.5)	97.7 (97.5)	57.0 (56.6)
R-base	94.9 (94.6)	**95.6** (94.9)	**98.3** (97.9)	97.8 (97.5)	**57.4** (56.9)
E-base	93.8 (93.0)	94.5 (93.5)	98.1 (97.9)	97.8 (97.5)	56.1 (55.7)
M-base	95.2 **(94.8)**	**95.6** (94.9)	98.2 **(98.0)**	97.7 (97.5)	**57.4 (57.1)**
E-large	95.2 (94.6)	95.3 (94.8)	98.2 **(97.8)**	97.8 (97.6)	57.2 (56.9)
R-large	**95.8** (94.9)	95.8 (94.9)	**98.3** (97.7)	97.8 (97.6)	58.8 (58.4)
M-large	95.7 **(95.0)**	**95.9 (95.1)**	98.1 **(97.8)**	**97.9 (97.7)**	**59.0 (58.8)**

The results are depicted in Table V and VI. Using additional pre-training data results in further improvement, as shown in the comparison between BERT-wwm and BERT-wwm-ext. This is why we use extended data for RoBERTa, ELECTRA, and MacBERT. Moreover, the proposed MacBERT yields significant improvements on all reading comprehension datasets. It is worth mentioning that our MacBERT-large could achieve a state-of-the-art F1 of 60% on the challenge set of CMRC 2018, which requires deeper text understanding.

Also, it should be noted that though DRCD is a traditional Chinese dataset, training with additional large-scale simplified Chinese could also have a great positive effect. As simplified and traditional Chinese share many identical characters, using a powerful pre-trained language model with only a few traditional

TABLE VIII
RESULTS ON SENTENCE PAIR CLASSIFICATION TASKS: XNLI, LCQMC, BQ CORPUS, AND OCNLI

	XNLI		LCQMC		BQ Corpus		OCNLI
	Dev	Test	Dev	Test	Dev	Test	Dev
BERT	77.8 (77.4)	77.8 (77.5)	89.4 (88.4)	86.9 (86.4)	86.0 (85.5)	84.8 (84.6)	74.6 (74.2)
BERT-wwm	79.0 (78.4)	78.2 (78.0)	89.4 (89.2)	87.0 (86.8)	86.1 **(85.6)**	85.2 **(84.9)**	74.6 (74.3)
BERT-wwm-ext	79.4 (78.6)	78.7 (78.3)	89.6 (89.2)	87.1 (86.6)	**86.4** (85.5)	**85.3** (84.8)	76.0 (75.3)
RoBERTa-wwm-ext	80.0 (79.2)	78.8 (78.3)	89.0 (88.7)	86.4 (86.1)	86.0 (85.4)	85.0 (84.6)	76.5 (76.0)
ELECTRA-base	77.9 (77.0)	78.4 (77.8)	**90.2 (89.8)**	**87.6 (87.3)**	84.8 (84.7)	84.5 (84.0)	76.1 (75.8)
MacBERT-base	**80.3 (79.7)**	**79.3 (78.8)**	89.5 (89.3)	87.0 (86.5)	86.0 (85.5)	85.2 **(84.9)**	**77.0 (76.5)**
ELECTRA-large	81.5 (80.8)	81.0 **(80.9)**	**90.7 (90.4)**	87.3 **(87.2)**	**86.7 (86.2)**	85.1 (84.8)	78.8 (78.4)
RoBERTa-wwm-ext-large	82.1 (81.3)	81.2 (80.6)	90.4 (90.0)	87.0 (86.8)	86.3 (85.7)	**85.8** (84.9)	78.5 (78.2)
MacBERT-large	**82.4 (81.8)**	**81.3** (80.6)	90.6 (90.3)	**87.6** (87.1)	86.2 **(85.7)**	85.6 **(85.0)**	**79.0 (78.7)**

Chinese data could also bring improvements without converting traditional Chinese characters into simplified ones.

Regarding CJRC, where the text is written in professional ways regarding Chinese laws, BERT-wwm shows moderate improvement over BERT but not that salient, indicating that further domain adaptation is needed for the fine-tuning tasks on non-general domains. However, increasing general pre-training data results in improvement, suggesting that when there is not enough domain data, we could also use large-scale general data as a remedy.

B. Single Sentence Classification

For the single sentence classification tasks, we select ChnSentiCorp, THUCNews, and TNEWS datasets. We use the ChnSentiCorp for evaluating sentiment classification, where the text should be classified into either a positive or negative label. THUCNews is a dataset that contains news in different genres, where the text is typically very long. In this paper, we use a version that contains 50K news in 10 domains (evenly distributed), including sports, finance, technology, etc.[7] TNEWS is a short text classification task consisting of news titles and keywords. TNEWS requires to classify into one of 15 classes. The results show that MacBERT could give moderate improvements over baselines in ChnSentiCorp and THUCNews, as these datasets have already reached high accuracies. In TNEWS, we can see that our MacBERT yields consistent improvements across base-level and large-level PLMs.

C. Sentence Pair Classification

For sentence pair classification tasks, we use XNLI data (Chinese portion), Large-scale Chinese Question Matching Corpus (LCQMC), BQ Corpus, and OCNLI, which require to input two sequences and predict their relations.

In XNLI and OCNLI, we can see that MacBERT yields relatively consistent and significant improvements over baselines. However, MacBERT only shows moderate improvements on LCQMC and BQ Corpus, with a slight improvement on the average score, but the peak performance is not as good as RoBERTa-wwm-ext-large. We suspect that these tasks are less sensitive to the subtle difference of the input than the reading comprehension tasks. As sentence pair classification only needs to generate a unified representation of the whole input and thus results in a moderate improvement.

We also noticed that the improvements are bigger in MRC tasks than classification tasks, while it might attribute to the masking strategy. In MRC tasks, the models should identify the exact answer span in the passage. In MacBERT, each word of N-gram is either replaced by its synonym or a random word, and thus each word can be easily identified, which forces the model to learn the word boundaries.

Another observation is that MacBERT-base generally yields larger improvements than MacBERT-large. This might be caused by two reasons. Firstly, MacBERT-base is initialized by BERT-base, which could benefit from the knowledge in BERT-base and avoid the cold-starting issue. Secondly, the results of large-level PLMs are generally higher than those of base-level PLMs, and thus getting a higher score is much difficult than base-level PLMs.

D. Results on Small Models

We also build a series of small models, namely RBT, built on either RoBERTa-base or RoBERTa-large models. The experimental results are shown in Table IX. Small models perform worse than the general models (base-level, large-level), because they use fewer parameters. As we can see that the performance drops in classification tasks are smaller than the reading comprehension tasks, indicating that it is possible to sacrifice minor performance to obtain a faster and smaller model, which could be beneficial for real-life applications. Also, by comparing RBTL3 and RBT6, which have similar parameter sizes, we can see that RBT6 substantially outperforms RBTL3, which indicates that a thin-and-tall model usually outperforms a wide-and-short model. These observations could be helpful in future model design for real-life applications.

VII. DISCUSSION

Based on the experimental results, we can see that these pre-trained language models also yield significant improvements over traditional BERT in Chinese tasks, indicating their effectiveness and generalizability. While our models achieve significant improvements on various Chinese tasks, we wonder where the essential components of the improvements from. To this end, we carry out detailed ablations on MacBERT to

[7] [Online]. Available: https://github.com/gaussic/text-classification-cnn-rnn

TABLE IX
RESULTS ON RBT SERIES, WHICH ARE BUILT ON ROBERTA-LARGE (ROBERTA-WWM-EXT-LARGE) AND ROBERTA-BASE (ROBERTA-WWM-EXT)

System	Params	CMRC 2018		DRCD		CJRC		CSC	THUC	XNLI	LC	BQ	AVG
		EM	F1	EM	F1	EM	F1	ACC	ACC	ACC	ACC	ACC	
RoBERTa-large	324M	74.2	90.6	89.6	94.5	62.4	82.2	95.8	97.8	81.2	87.0	85.8	86.79
RoBERTa-base	102M	72.6	89.4	85.6	92.0	59.0	79.0	95.6	97.8	78.8	86.4	85.0	85.30
RBTL3	61M	63.3	83.4	77.2	85.6	64.6	74.9	94.2	97.8	74.0	85.1	83.6	82.40
RBT3	38M	62.2	81.8	75.0	83.9	63.5	73.7	92.8	97.5	72.3	85.1	83.3	81.38
RBT4	45M	65.0	83.9	78.7	86.7	65.5	75.3	93.8	97.7	74.2	85.7	83.7	82.83
RBT6	60M	68.3	84.4	83.9	90.2	69.1	78.8	95.3	97.8	76.2	86.6	84.2	84.68

TABLE X
ABLATIONS OF MACBERT-LARGE ON DIFFERENT FINE-TUNING TASKS

System	CMRC 2018		DRCD		CJRC		CSC	THUC	XNLI	LC	BQ	AVG
	EM	F1	EM	F1	EM	F1	ACC	ACC	ACC	ACC	ACC	
MacBERT-large	74.8	90.7	91.7	95.6	62.9	82.5	95.9	97.9	81.3	87.6	85.6	87.18
SOP $\rightarrow$ NSP	74.5	90.6	91.5	95.5	62.4	82.3	96.0	97.8	81.2	87.4	85.2	87.00
w/o SOP	74.4	90.6	91.0	95.4	62.2	82.1	95.8	97.8	81.1	87.4	85.2	86.89
w/o Mac	74.2	90.1	91.2	95.4	62.2	82.3	95.7	97.8	81.2	87.4	85.3	86.88
w/o NM	74.0	89.8	90.9	95.1	62.1	82.0	95.9	97.9	81.3	87.5	85.6	86.89
RoBERTa-large	74.2	90.6	89.6	94.5	62.4	82.2	95.8	97.8	81.2	87.0	85.8	86.79

demonstrate its effectiveness, and we also compare the claims of the existing pre-trained language models in English to see if their modification still holds true in another language.

A. Effectiveness of MacBERT

We carry out detailed ablations to examine the contributions of each component in MacBERT. The results are shown in Table X.

The overall average scores are obtained by averaging the test scores of each task (EM and F1 metrics are averaged before the overall averaging). From a general view, removing any component in MacBERT results in a decline in the average performance, suggesting that all modifications contribute to the overall improvements. Specifically, the most effective modifications are the N-gram masking and similar word replacement, which are the modifications on the masked language model task. When we compare N-gram masking and similar word replacement, we could see clear pros and cons, where N-gram masking seems to be more effective in text classification tasks, and the performance of reading comprehension tasks seems to benefit more from the similar word replacement task. Combining these two tasks could compensate for each other and have a better performance on both genres.

The NSP task does not show as much importance as the MLM task, demonstrating that it is much more important to design a better MLM task to fully unleash the text modeling power. Also, we compared the next sentence prediction [2] and sentence order prediction [15] task to better judge which one is much powerful. The results show that the sentence order prediction task indeed shows better performance than the original NSP, though it is not that salient. The SOP task requires identifying the correct order of the two sentences rather than using a random sentence, which is much easy for the machine to identify. Removing the SOP task results in noticeable declines in reading comprehension tasks compared to the text classification tasks, which suggests that it is necessary to design an NSP-like task to learn the relations between two segments (for example, passage and question in reading comprehension task).

B. Investigation on MLM Task

As illustrated in the previous section, the dominant pre-training task is the masked language model and its variants. The masked language model task relies on two sides: 1) the selection of the tokens to be masked, and 2) the replacement of the selected tokens. In the previous section, we have demonstrated the effectiveness of the selection of the masking tokens, such as the whole word masking or N-gram masking, etc. Now we are going to investigate how the replacement of the selected tokens affects the performance of the pre-trained language models. In order to investigate this problem, we plot the CMRC 2018 and DRCD performance at different pre-training steps. Specifically, we follow the original masking percentage 15% of the input sequence, in which 10% masked tokens remain the same. In terms of the remaining 90% masked tokens, we classify them into four categories.

- *MacBERT:* 80% tokens replaced into their similar words, and 10% replaced into random words.
- *Random Replace:* 90% tokens replaced into random words.
- *Partial Mask:* original BERT implementation, with 80% tokens replaced into `[MASK]` tokens, and 10% replaced into random words.
- *All Mask:* 90% tokens replaced with `[MASK]` tokens.

We only plot the steps from 1M to 2M to show stabler results than the first 1M steps. The results are depicted in Fig. 2.

The pre-training models that rely on mostly using `[MASK]` for masking purposes (i.e., partial mask and all mask) result in worse performances, indicating that the discrepancy of the pre-training and fine-tuning is an actual problem that affects the overall performance. Among which, we also noticed that if we do not leave 10% as original tokens (i.e., identity projection), there is also a consistent decline, indicating that masking with

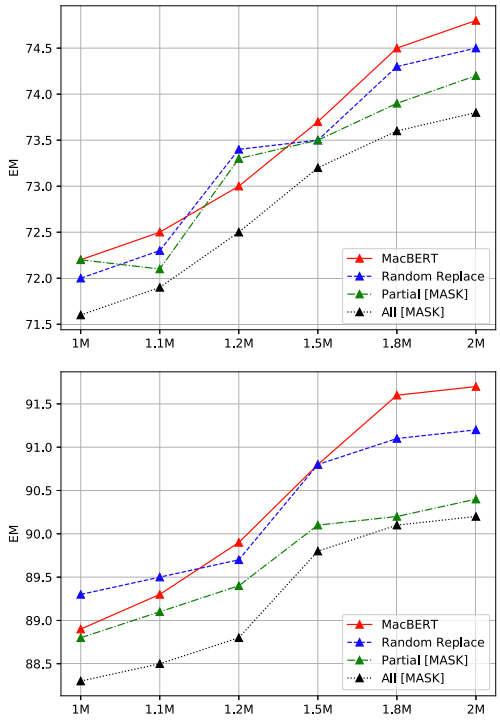
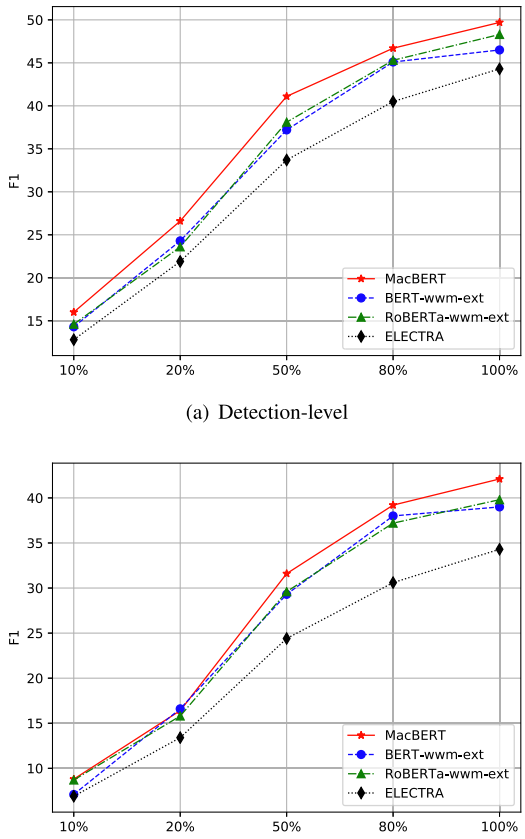


Fig. 2. Results of different MLM tasks on CMRC 2018 and DRCD.



(a) Detection-level

(b) Correction-level

Fig. 3. Results of using different percentage of SIGHAN-15 training data.

[MASK] token is less robust and vulnerable to the absence of identity projection for negative sample training.

To our surprise, a quick fix, that is to abandon the [MASK] token completely and replace all 90% masked tokens into random words, yields consistent improvements over [MASK]-dependent masking strategies. This also strengthens the claims that the original masking method that relies on the [MASK] token, which never appears in the fine-tuning task, resulting in a discrepancy and worse performance. Also, using random words rather than the artificial token [MASK] could improve the de-noising ability of the pre-trained model, which might also be a possible reason. To make this more delicate, in this paper, we propose to use similar words for masking purposes, instead of randomly pick a word from the vocabulary, as random words are not fit in the context and may break the naturalness of the language model learning, as traditional N-gram language model is based on natural sentence rather than a manipulated influent sentence. However, if we use similar words for masking purposes, the fluency of the sentence is much better than using random words, and the whole task transforms into a grammar correction task, which is much more natural and without the discrepancy of the pre-training and fine-tuning stage. From the figure, we can see that the MacBERT yields the best performance among the four variants, which verifies our assumptions.

C. Analyses on Chinese Spell Check

MacBERT introduces 'MLM as correction' tasks, which is similar to the actual grammar or spell error correction tasks. We perform additional experiments on Chinese Spell Check tasks. We use SIGHAN-15 [40] dataset to explore the effect of different pre-trained language models when using different percentages of training data. SIGHAN-15 consists of a training set of 3.1K instances and a test set of 1.1K instances. We compare BERT-wwm-ext, RoBERTa-wwm-ext, ELECTRA-base, and MacBERT-base in this experiment, as they share the same pre-training data. We fine-tune each model five times and plot the figures with averaged F1 (sentence-level). We use a universal learning rate of 5e-5 and train 5 epochs with a batch size of 64. The results are shown in Fig. 3, including detection-level and correction-level scores.

As we can see that our MacBERT yields consistent improvements over others when using different percentages of the training data, indicating that our approach is effective and scalable. We notice that ELECTRA does not perform well on this task. Especially, the gap between ELECTRA and others on the correction-level results are relatively larger than that in the detection-level. ELECTRA uses replaced token detection (RTD) task for training the discriminator (which will be used for fine-tuning). However, the RTD task only needs to identify whether the input tokens are altered without predicting the original token, which we think is quite simple. On the contrary, MLM and Mac objectives require identify-and-correction at the same time. By comparing MLM and Mac, our MacBERT alleviates the discrepancy of pre-training and fine-tuning issues, which yields another significant gain.

We note that though the Mac task is similar to the spell check task, we only use synonyms for replacement, which is only a small proportion in real spell check tasks. This could explain why our model does not yield larger improvement over others when there is fewer training data available.

VIII. Conclusion

In this paper, we revisit pre-trained language models in Chinese to see if the techniques in these state-of-the-art models generalize well in a different language other than English only. We created Chinese pre-trained language model series and proposed a new model called MacBERT, which modifies the masked language model (MLM) task as a language correction manner and mitigates the discrepancy of the pre-training and fine-tuning stage. Extensive experiments are conducted on various Chinese NLP datasets, and the results show that the proposed MacBERT could give significant gains in most of the tasks, and detailed ablations show that more focus should be made on the MLM task rather than the NSP task and its variants, as we found that NSP-like task does not show a landslide advantage over one another. With the release of the Chinese pre-trained language model series, we hope it will further accelerate the natural language processing in our research community.

In the future, we would like to investigate an effective way to determine the masking ratios instead of heuristic ones to further improve the performance of the pre-trained language models. Also, we would like to design more effective language modeling approaches to further exploit large-scale unsupervised data.

Acknowledgment

The authors would like to thank all anonymous reviewers and editors for their thorough reviewing and providing constructive comments to improve our paper.

References

[1] Y. Cui *et al.*, "Pre-training with whole word masking for Chinese BERT," 2019, *arXiv:1906.08101*.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[3] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 784–789. [Online]. Available: https://www.aclweb.org/anthology/P18-2124

[4] S. Reddy, D. Chen, and C. D. Manning, "CoQA: A conversational question answering challenge," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 249–266, 2019.

[5] E. Choi *et al.*, "QuAC: Question answering in context," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2174–2184. [Online]. Available: https://www.aclweb.org/anthology/D18-1241

[6] T. Kwiatkowski *et al.*, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 453–466, 2019.

[7] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale reading comprehension dataset from examinations," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Association for Computational Linguistics, 2017, pp. 796–805. [Online]. Available: http://www.aclweb.org/anthology/D17-1083

[8] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 2978–2988. [Online]. Available: https://www.aclweb.org/anthology/P19-1285

[9] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, and X. Zhou, "DCMN+: Dual co-matching network for multi-choice reading comprehension," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 5, Apr. 2020, pp. 9563–9570.

[10] Q. Ran, P. Li, W. Hu, and J. Zhou, "Option comparison network for multiple-choice reading comprehension," 2019, *arXiv:1903.03033*.

[11] Y. Sun *et al.*, "ERNIR: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*.

[12] Z. Yang *et al.*, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach *et al.* Eds., Curran Associates, Inc., vol. 32, 2019, pp. 1–11.

[13] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: https://www.cs.princeton.edu/ danqic/papers/roberta_paper.pdf

[14] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, pp. 64–77, 2020.

[15] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–17. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[16] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–18. [Online]. Available: https://openreview.net/pdf?id=r1xMH1BtvB

[17] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[18] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[19] L. Kong, C. de M. d'Autume, L. Yu, W. Ling, Z. Dai, and D. Yogatama, "A mutual information maximization perspective of language representation learning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=Syx79eBKwr

[20] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[21] W. Che, Z. Li, and T. Liu, "LTP: A chinese language technology platform," in *Proc. 23rd Int. Conf. Comput. Linguistics. Demonstrations*. Association for Computational Linguistics, 2010, pp. 13–16.

[22] Y. Levine *et al.*, "PMI-masking: Principled masking of correlated spans," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1-14. [Online]. Available: https://openreview.net/forum?id=3Aoft6NWFej

[23] T. Mikolov, I. Sutskever, K. G. S. Chen Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems* 26, C. J. C. L. Burges, M. Bottou, Z. Welling Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013, pp. 3111–3119.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980.

[25] Y. You, J. Li, J. Hseu, X. Song, J. Demmel, and C.-J. Hsieh, "Reducing BERT pre-training time from 3 days to 76 minutes," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–38. [Online]. Available: https://openreview.net/forum?id=Syx4wnEtvH

[26] Y. Cui *et al.*, "A span-extraction dataset for Chinese machine reading comprehension," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5886–5891. [Online]. Available: https://www.aclweb.org/anthology/D19-1600

[27] C. C. Shao, T. Liu, Y. Lai, Y. Tseng, and S. Tsai, "DRCD: A Chinese machine reading comprehension dataset," 2018, *arXiv:1806.00920*.

[28] X. Duan *et al.*, "CJRC: A reliable human-annotated benchmark dataset for Chinese judicial reading comprehension," in *Proc. China Nat. Conf. Chin. Comput. Linguistics*, Springer, 2019, pp. 439–451.

[29] S. Tan and J. Zhang, "An empirical study of sentiment analysis for Chinese documents," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2622–2629, 2008.

[30] J. Li and M. Sun, "Scalable term selection for text categorization," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007, pp. 774–782.

[31] L. Xu *et al.*, "CLUE: A Chinese language understanding evaluation benchmark," in *Proc. 28th Int. Conf. Comput. Linguistics*, Barcelona, Spain: International Committee on Computational Linguistics, 2020, pp. 4762–4772. [Online]. Available: https://aclanthology.org/2020.coling-main.419
[32] A. Conneau *et al.*, "XNLI: Evaluating cross-lingual sentence representations," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Association for Computational Linguistics, 2018, pp. 2475–2485.
[33] X. Liu *et al.*, "LCQMC: A large-scale Chinese question matching corpus," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1952–1962.
[34] J. Chen, Q. Chen, X. Liu, H. Yang, D. Lu, and B. Tang, "The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4946–4951. [Online]. Available: https://www.aclweb.org/anthology/D18-1536
[35] H. Hu, K. Richardson, L. Xu, L. Li, S. Kuebler, and L. Moss, "OCNLI: Original Chinese natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process., Findings*, 2020, pp. 3512–3526. [Online]. Available: https://arxiv.org/abs/2010.05444
[36] Y. Sun *et al.*, "ERNIE 2.0: A continual pre-training framework for language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8968–8975.
[37] J. Wei *et al.*, "NEZHA: Neural contextualized representation for Chinese language understanding," Aug. 2019, *arXiv:1909.00204*.
[38] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Des. Implementation*, 2016, pp. 265–283.
[39] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000 questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Association for Computational Linguistics, 2016, pp. 2383–2392. [Online]. Available: http://www.aclweb.org/anthology/D16-1264
[40] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to SIGHAN 2015 bake-off for Chinese spelling check," in *Proc. 8th SIGHAN Workshop Chin. Lang. Process.*, Beijing, China: Association for Computational Linguistics, 2015, pp. 32–37. [Online]. Available: https://aclanthology.org/W15-3106

Yiming Cui received the M.S. and B.S. degrees and is currently working toward the Doctoral degree with the Harbin Institute of Technology, Harbin, China. He is the Principal Researcher with the Joint Laboratory of HIT and iFLYTEK Research (HFL).

His main research interests include machine reading comprehension, question answering, and pre-trained language model, etc. He has authored or coauthored more than 20 papers in top conferences, such as in ACL, EMNLP, AAAI, COLING, NAACL, etc. He is a Senior Member of China Computer Federation (CCF).

Wanxiang Che is a Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He was the Vice Director of Research Center for Social Computing and Information Retrieval. He is a Young Scholar of "Heilongjiang Scholar" and a Visiting Scholar of Stanford University. He is currently the Vice Director and Secretary-General of the Computational Linguistics Professional Committee of CIPS and a CCF Senior Member. He achieved the AAAI 2013 Outstanding Paper Honorable Mention Award. His Language Technology Platform (LTP) has been shared by more than 600 organizations and authorized to Baidu, Tencent, and so on.

Ting Liu received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 1998. He is a Full Professor and the Director of the Department of Computer Science at Harbin Institute of Technology. He has authored or coauthored hundreds of papers with more than 20,000 citations.

His research interests include information retrieval, natural language processing, and social media analysis.

Bing Qin is a Full Professor and Doctoral Supervisor with the School of Computer Science, Harbin Institute of Technology, Harbin, China. She is also the Director with Research Center for Social Computing and Information Retrieval (HIT-SCIR), Harbin Institute of Technology. She has authored or coauthored more than 80 papers in top conferences, such as ACL, COLING, EMNLP, IEEE TKDE, IEEE TASLP, etc. Her main research interests include natural language processing, information extraction, text mining, emotion analysis, etc.

Ziqing Yang received the B.S degree from Wuhan University, Wuhan, China, in 2010 and the Ph.D. degree in physics from the University of Chinese Academy of Sciences, China, in 2017. He is a Researcher with the Joint Laboratory of HIT and iFLYTEK Research (HFL). He has authored or coauthored several top-tier conference papers, including ACL, COLING, etc. He has a broad interest in machine learning and natural language processing, including machine reading comprehension, knowledge distillation for NLP and general machine learning method for NLP.