Revisiting Pre-trained Models for Chinese Natural Language Processing

Yiming Cui[1,2], **Wanxiang Che**[1], **Ting Liu**[1], **Bing Qin**[1], **Shijin Wang**[2,3], **Guoping Hu**[2]
[1]Research Center for Social Computing and Information Retrieval (SCIR),
Harbin Institute of Technology, Harbin, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
[3]iFLYTEK AI Research (Hebei), Langfang, China
[1]{ymcui,car,tliu,qinb}@ir.hit.edu.cn
[2,3]{ymcui,sjwang3,gphu}@iflytek.com

Abstract

Bidirectional Encoder Representations from Transformers (BERT) has shown marvelous improvements across various NLP tasks, and consecutive variants have been proposed to further improve the performance of the pre-trained language models. In this paper, we target on revisiting Chinese pre-trained language models to examine their effectiveness in a non-English language and release the Chinese pre-trained language model series to the community. We also propose a simple but effective model called MacBERT, which improves upon RoBERTa in several ways, especially the masking strategy that adopts *MLM as correction* (Mac). We carried out extensive experiments on eight Chinese NLP tasks to revisit the existing 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.[1]

1 Introduction

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) 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 (Rajpurkar et al., 2018), CoQA (Reddy et al., 2019), QuAC (Choi et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019), RACE (Lai et al., 2017), we can see that most of the top-performing models are based on BERT and its variants (Dai et al., 2019; Zhang et al., 2019; Ran et al., 2019), demonstrating that the pre-trained language models have become new fundamental components in natural language processing field.

Starting from BERT, the community have made great and rapid progress on optimizing the pre-trained language models, such as ERNIE (Sun et al., 2019a), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2019), ALBERT (Lan et al., 2019), ELECTRA (Clark et al., 2020), etc. However, training Transformer-based (Vaswani et al., 2017) pre-trained language models are not as easy as we used to train word embeddings or other traditional neural networks. Typically, training a powerful BERT-large model, which has 24-layer Transformer with 330 million parameters, to convergence needs high-memory computing devices, such as TPU, which is 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 on other languages.

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 if these models generalize well in the 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 and mitigates the discrepancy of the pre-training and fine-tuning stage. Extensive experiments are conducted on eight popular Chinese NLP datasets, ranging from sentence-level to document-level, 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

[1] https://github.com/ymcui/MacBERT

	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

Table 1: Comparisons of the pre-trained language models. (AE: Auto-Encoder, AR: Auto-Regressive, T: Token, S: Segment, P: Position, W: Word, 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, Mac: MLM as correction)

models, and detailed ablations are given to better examine the composition of the improvements. The contributions of this paper are listed as follows.

- Extensive empirical studies are carried out to revisit the performance of Chinese 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 down-stream tasks.

- To further accelerate future research on Chinese NLP, we create and release the Chinese pre-trained language model series to the community.

2 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 1. We elaborate on their key components in the following subsections.

2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) has proven to be successful in natural language processing studies. 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 A.

Later, they further proposed a technique called whole word masking (wwm) for optimizing the original masking in the MLM task. In this setting, instead of randomly selecting WordPiece (Wu et al., 2016) tokens to mask, we always mask all of the tokens corresponding to a whole word at once. This will explicitly force the model to recover the whole word in the MLM pre-training task instead of just recovering WordPiece tokens (Cui et al., 2019a), 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 down-stream 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 stimulates the community to have great interests in the research of pre-trained language models.

2.2 ERNIE

ERNIE (Enhanced Representation through kNowledge IntEgration) (Sun et al., 2019a) 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 will mask 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 (Devlin et al., 2019; Joshi et al., 2019).[2].

[2] Though N-gram masking was not included in Devlin et al. (2019), according to their model name in the SQuAD leaderboard, we often admit their credit towards this method.

2.3 XLNet

Yang et al. (2019) argues that the existing pre-trained language models that are based on autoencoding, such as BERT, suffer from the discrepancy of the pre-training and fine-tuning stage because the masking symbol `[MASK]` will never appear in the fine-tuning stage. To alleviate this problem, they proposed XLNet, which was based on Transformer-XL (Dai et al., 2019). 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 called the Permutation Language Model (PLM). Another is to change the autoencoding language model into an autoregressive one, which is similar to the traditional statistical language models.[3]

2.4 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019) aims to adopt original BERT architecture but make much more precise modifications to show the powerfulness of BERT, which was underestimated. They carried out careful comparisons of various components in BERT, including the masking strategies, training steps, etc. After thorough evaluations, they came 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 and using dynamic masking.

2.5 ALBERT

ALBERT (A Lite BERT) (Lan et al., 2019) primarily tackles the problems of higher memory assumption and slow training speed of BERT. ALBERT introduces two parameter reduction techniques. The first one is the factorized embedding parameterization that 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 will significantly reduce the parameters. Besides, they also proposed the sentence-order prediction (SOP) task to replace the traditional NSP pre-training task.

2.6 ELECTRA

ELECTRA (Efficiently Learning an Encoder that Classifiers Token Replacements Accurately) (Clark et al., 2020) employs a new generator-discriminator framework that is similar to GAN (Goodfellow et al., 2014). 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. 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. In the fine-tuning stage, only the discriminator is used.

3 Chinese Pre-trained Language Models

While we believe most of the conclusions in the previous works are true in English condition, we wonder if these techniques still generalize well in other languages. In this section, we illustrate how the existing pre-trained language models are adapted for the Chinese language. Furthermore, we also propose a new model called MacBERT, which adopts the advantages of previous models as well as newly designed components. Note that, as these models are all originated from BERT 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.

3.1 BERT-wwm & RoBERTa-wwm

In the original BERT, a WordPiece tokenizer (Wu et al., 2016) was used to split the text into WordPiece tokens, where some words will be split into several small fragments. The whole word masking (wwm) mitigate 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 split 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 whole word masking in Chinese to mask the word instead of individual Chinese characters. For implementation, we strictly followed the original whole word masking codes and did not change other components, such as the percentage of word masking, etc. We use LTP (Che et al., 2010) for Chinese word segmentation to identify the word boundaries.

[3] We also trained Chinese XLNet, but it only shows competitive performance on reading comprehension datasets. We've included these results in the Appendix.

	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	使用 语言 模型 来 预测 下 一个 词 的 概率 。	-
+ BERT Tokenizer	使 用 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	使 用 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	使 用 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	使 用 **[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	使 用 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 .

Figure 1: Examples of different masking strategies.

Note that the whole word masking only affects the selection of the masking tokens in the pre-training stage. The input of BERT still uses WordPiece tokenizer to split the text, which is identical to the original BERT.

Similarly, whole word masking could also be applied on RoBERTa, where the NSP task is not adopted. An example of the whole word masking is depicted in Figure 1.

3.2 MacBERT

In this paper, we take advantage of previous models and propose a simple modification that leads to significant improvements on fine-tuning tasks, where we call this model as **MacBERT** (**M**LM **a**s **c**orrection **BERT**). MacBERT shares the same pre-training tasks as BERT with several modifications. For the MLM task, we perform the following modifications.

- We use whole word masking as well as N-gram masking strategies for selecting candidate tokens for masking, with a percentage of 40%, 30%, 20%, 10% for word-level unigram to 4-gram.

- 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 (Wang and Hu, 2017), which is based on word2vec (Mikolov et al., 2013) similarity calculations. If an N-gram is selected to mask, we will find similar words individually. In rare cases, when there is no similar word, we will degrade to use random word replacement.

- We use a percentage of 15% input words for masking, where 80% will replace with similar words, 10% replace with a random word, and keep with original words for the rest of 10%.

For the NSP-like task, we perform sentence-order prediction (SOP) task as introduced by ALBERT (Lan et al., 2019), where the negative samples are created by switching the original order of two consecutive sentences. We ablate these modifications in Section 6.1 to better demonstrate the contributions of each component.

4 Experimental Setups

4.1 Setups for Pre-Trained Language Models

We downloaded Wikipedia dump[4] (as of March 25, 2019), and pre-processed with `WikiExtractor.py` as suggested by Devlin et al. (2019), resulting in 1,307 extracted files. We use both Simplified and Traditional Chinese in this dump. After cleaning the raw text (such as removing `html` tagger) 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, we use LTP (Che et al., 2010) for Chinese word segmentation. We use official `create_pretraining_data.py` to convert raw input text to the pre-training examples.

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 using the same vocabulary provided by the base-level model.

For training BERT series, we adopt the scheme of training on a maximum length of 128 tokens then on 512, suggested by Devlin et al. (2019). However, we empirically found that this will result in

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

Task	Dataset	Domain	MaxLen	Batch	Epoch	InitLR	Train #	Dev #	Test #
MRC	CMRC 2018	Wikipedia	512	64	2	3e-5	10K	3.2K	4.9K
	DRCD	Wikipedia	512	64	2	3e-5	27K	3.5K	3.5K
	CJRC	Law	512	64	2	4e-5	10K	3.2K	3.2K
SSC	ChnSentiCorp	Various	256	64	3	2e-5	9.6K	1.2K	1.2K
	THUCNews	News	512	64	3	2e-5	50K	5K	10K
SPC	XNLI	Various	128	64	2	3e-5	392K	2.5K	5K
	LCQMC	Zhidao	128	64	3	2e-5	240K	8.8K	12.5K
	BQ Corpus	QA	128	64	3	3e-5	100K	10K	10K

Table 2: Data statistics and hyper-parameter settings for different fine-tuning tasks.

	BERT	BERT-wwm	RoBERTa-wwm	ELECTRA	MacBERT
Word #	0.4B	5.4B	5.4B	5.4B	5.4B
Vocab #	21,128	21,128	21,128	21,128	21,128
Hidden Activation	GeLU	GeLU	GeLU	GeLU	GeLU
Optimizer	AdamW	LAMB	AdamW	AdamW	LAMB
Training Steps	?	2M	1M	2M	1M
Init Checkpoint	random	BERT	BERT	random	BERT

Table 3: Training details of Chinese pre-trained language models.

insufficient adaptation for the long-sequence tasks, such as reading comprehension. In this context, for RoBERTa and MacBERT, we directly use a maximum length of 512 throughout the pre-training process, which was adopted in Liu et al. (2019). For the batch size less than 1024, we adopt the original ADAM (Kingma and Ba, 2014) with weight decay optimizer in BERT for optimization, and use LAMB optimizer (You et al., 2019) for better scalability in larger batch. 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 3. For clarity, we do not list '`ext`' models, where the other parameters are the same with the one that is not trained on extended data.

4.2 Setups for Fine-tuning Tasks

To thoroughly test these pre-trained language models, we carried 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 2. Specifically, we choose the following eight popular Chinese datasets.

- **Machine Reading Comprehension (MRC)**: CMRC 2018 (Cui et al., 2019b), DRCD (Shao et al., 2018), CJRC (Duan et al., 2019).

- **Single Sentence Classification (SSC)**: ChnSentiCorp (Tan and Zhang, 2008), THUCNews (Li and Sun, 2007).

- **Sentence Pair Classification (SPC)**: XNLI (Conneau et al., 2018), LCQMC (Liu et al., 2018), BQ Corpus (Chen et al., 2018).

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 original Chinese BERT, and it would be possible to achieve another gains by tuning the learning rate individually. We run the same experiment ten times to ensure the reliability of 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.

For all models except for ELECTRA, we use the same initial learning rate setting for each task, as depicted in Table 2. 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 Clark et al. (2020).

As the pre-training data are quite different

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

CMRC 2018	**Dev**		**Test**		**Challenge**	
	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)
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)
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)
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)
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)
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)**
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)
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)
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)**

Table 4: Results on CMRC 2018 (Simplified Chinese). 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).

among various existing Chinese pre-trained language models, such as ERNIE (Sun et al., 2019a), ERNIE 2.0 (Sun et al., 2019b), NEZHA (Wei et al., 2019), we only compare BERT (Devlin et al., 2019), 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 by Devlin et al. (2019). We carried out experiments under TensorFlow framework (Abadi et al., 2016) with slight modifications to the fine-tuning scripts[6] provided by Devlin et al. (2019) to better adapt to Chinese.

5 Results

5.1 Machine Reading Comprehension

Machine Reading Comprehension (MRC) is a representative document-level modeling task which 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 (Rajpurkar et al., 2016) that extract a passage span for the given question.

- **DRCD**: This is also a span-extraction MRC dataset but in Traditional Chinese.

- **CJRC**: Similar to CoQA (Reddy et al., 2019), 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.

DRCD	**Dev**		**Test**	
	EM	**F1**	**EM**	**F1**
BERT	83.1 (82.7)	89.9 (89.6)	82.2 (81.6)	89.2 (88.8)
BERT-wwm	84.3 (83.4)	90.5 (90.2)	82.8 (81.8)	89.7 (89.0)
BERT-wwm-ext	85.0 (84.5)	91.2 (90.9)	83.6 (83.0)	90.4 (89.9)
RoBERTa-wwm-ext	86.6 (85.9)	92.5 (92.2)	85.6 (85.2)	92.0 (91.7)
ELECTRA-base	87.5 (87.0)	92.5 (92.3)	86.9 (86.6)	91.8 (91.7)
MacBERT-base	**89.4 (89.2)**	**94.3 (94.1)**	**89.5 (88.7)**	**93.8 (93.5)**
ELECTRA-large	88.8 (88.7)	93.3 (93.2)	88.8 (88.2)	93.6 (93.2)
RoBERTa-wwm-ext-L	89.6 (89.1)	94.8 (94.4)	89.6 (88.9)	94.5 (94.1)
MacBERT-large	**91.2 (90.8)**	**95.6 (95.3)**	**91.7 (90.9)**	**95.6 (95.3)**

Table 5: Results on DRCD (Traditional Chinese).

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 6: Results on CJRC.

The results are depicted in Table 4, 5, 6. Using additional pre-training data will result 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

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

Single Sentence	**ChnSentiCorp**		**THUCNews**	
Classification	**Dev**	**Test**	**Dev**	**Test**
BERT	94.7 (94.3)	95.0 (94.7)	97.7 (97.4)	**97.8** **(97.6)**
BERT-wwm	95.1 (94.5)	95.4 **(95.0)**	98.0 (97.6)	**97.8** **(97.6)**
BERT-wwm-ext	**95.4** (94.6)	95.3 (94.8)	97.7 (97.5)	97.7 (97.5)
RoBERTa-wwm-ext	94.9 (94.6)	**95.6** (94.9)	**98.3** (97.9)	97.8 (97.5)
ELECTRA-base	93.8 (93.0)	94.5 (93.5)	98.1 (97.9)	97.8 (97.5)
MacBERT-base	95.2 **(94.8)**	**95.6** (94.9)	98.2 **(98.0)**	97.7 (97.5)
ELECTRA-large	95.2 (94.6)	95.3 (94.8)	98.2 **(97.8)**	97.8 (97.6)
RoBERTa-wwm-ext-large	**95.8** (94.9)	95.8 (94.9)	**98.3** (97.7)	97.8 (97.6)
MacBERT-large	95.7 **(95.0)**	**95.9** **(95.1)**	98.1 **(97.8)**	**97.9** **(97.7)**

Table 7: Results on single sentence classification tasks: ChnSentiCorp and THUCNews.

Sentence Pair	**XNLI**		**LCQMC**		**BQ Corpus**	
Classification	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
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)
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)**
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)
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)
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)
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)**
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)
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)
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)**

Table 8: Results on sentence pair classification tasks: XNLI, LCQMC, and BQ Corpus.

powerful pre-trained language model with only a few traditional 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, by increasing general training data will result in improvement, suggesting that when there is no enough domain data, we could also use large-scale general data as a remedy.

5.2 Single Sentence Classification

For single sentence classification tasks, we select ChnSentiCorp and THUCNews datasets. We use the ChnSentiCorp dataset 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] The results show that our MacBERT could give moderate improvements over baselines, as these datasets have already reached very high accuracies.

5.3 Sentence Pair Classification

For sentence pair classification tasks, we use XNLI data (Chinese portion), Large-scale Chinese Question Matching Corpus (LCQMC), and BQ Corpus, which require to input two sequences and predict their relations. We can see that MacBERT outperforms other models, but the improvements were moderate, 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.

6 Discussion

While our models achieve significant improvements on various Chinese tasks, we wonder where the essential components of the improvements from. To this end, we carried out detailed ablations on MacBERT to demonstrate their effectiveness,

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

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.

6.1 Effectiveness of MacBERT

We carried out ablations to examine the contributions of each component in MacBERT, which was thoroughly evaluated in all fine-tuning tasks. The results are shown in Table 9. 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 will result 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. By combining these two tasks will compensate 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 (Devlin et al., 2019) and sentence order prediction (Lan et al., 2019) 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 will result 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).

6.2 Investigation on MLM Task

As said 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 will affect 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 into four categories.

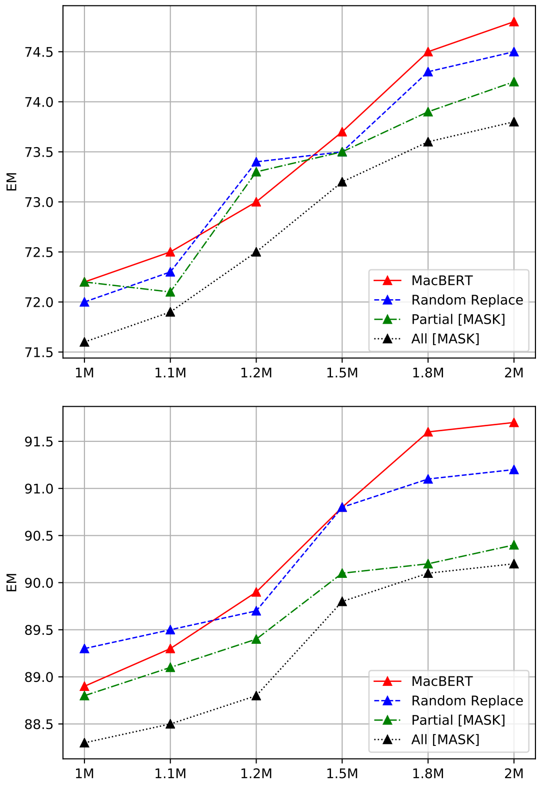

Figure 2: Results of different MLM tasks on CMRC 2018 and DRCD.

- **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.

	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 → 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

Table 9: Ablations of MacBERT-large on different fine-tuning tasks.

- **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 Figure 2. The pre-training models that rely on mostly using [MASK] for masking purpose (i.e., partial mask and all mask) results 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 [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, will result in a discrepancy and worse performance. To make this more delicate, in this paper, we propose to use similar words for masking purpose, instead of randomly pick a word from the vocabulary, as random word will 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 chart, we can see that the MacBERT yields the best performance among the four variants, which verifies our assumptions.

7 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 the Chinese 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.

Acknowledgments

We would like to thank all anonymous reviewers and senior program members for their thorough reviewing and providing constructive comments to improve our paper. The first author was partially supported by the Google TensorFlow Research Cloud (TFRC) program for Cloud TPU access. This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011, and 61772153.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 13–16. Association for Computational Linguistics.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4946–4951, Brussels, Belgium. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019a. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019b. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5886–5891, Hong Kong, China. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, et al. 2019. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. In *China National Conference on Chinese Computational Linguistics*, pages 439–451. Springer.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 796–805. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jingyang Li and Maosong Sun. 2007. Scalable term selection for text categorization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

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 arXiv:1907.11692*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Qiu Ran, Peng Li, Weiwei Hu, and Jie Zhou. 2019. Option comparison network for multiple-choice reading comprehension. *arXiv preprint arXiv:1903.03033*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

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

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019b. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Songbo Tan and Jin Zhang. 2008. An empirical study of sentiment analysis for chinese documents. *Expert Systems with applications*, 34(4):2622–2629.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Hailiang Wang and Yingxi Hu. 2017. Synonyms.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. NEZHA: Neural Contextualized Representation for Chinese Language Understanding. *arXiv e-prints*, page arXiv:1909.00204.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yang You, Jing Li, Jonathan Hseu, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. 2019. Reducing bert pre-training time from 3 days to 76 minutes. *arXiv preprint arXiv:1904.00962*.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dual co-matching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.

A Appendix

A.1 XLNet Results on Machine Reading Comprehension Tasks

Following official XLNet implementation, we trained a sentencepiece vocabulary of 32,000 and used it for word segmentation. We use exactly the same pre-training data as those marked as 'ext' models with 5.4B training tokens. We mainly implemented XLNet-base (12-layers, 768 hidden dimension) and XLNet-mid (24-layers, 768 hidden dimension). The pre-training of XLNet-mid and XLNet-base was done on a single Cloud TPU v3 for 2M/4M steps with a batch size of 32, respectively.

The results on CMRC 2018 and DRCD are shown in Table 10 and 11. The results show that these XLNet models could achieve moderate improvements over BERT, and the improvements are not consistent on each subset.

CMRC 2018	**Dev**		**Test**		**Challenge**	
	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)
XLNet-base	65.2 (63.0)	86.9 (85.9)	67.0 (65.8)	87.2 (86.8)	25.0 (22.7)	51.3 (49.5)
XLNet-mid	66.8 (66.3)	88.4 (88.1)	69.3 (68.5)	89.2 (88.8)	29.1 (27.1)	55.8 (54.9)

Table 10: Results of XLNet on CMRC 2018 (Simplified Chinese).

DRCD	**Dev**		**Test**	
	EM	**F1**	**EM**	**F1**
BERT	83.1 (82.7)	89.9 (89.6)	82.2 (81.6)	89.2 (88.8)
XLNet-base	83.8 (83.2)	92.3 (92.0)	83.5 (82.8)	92.2 (91.8)
XLNet-mid	85.0 (84.5)	91.2 (90.9)	85.5 (84.8)	93.6 (93.2)

Table 11: Results of XLNet on DRCD (Traditional Chinese).

We also carried out experiments on text classification task, such as XNLI, but the XLNet-mid could only gives near 74% on the test set, while the BERT-base could reach an accuracy of 77.8%. We haven't figured out the exact issues and also did not find other successful Chinese XLNet in the community. We will investigate the issue and will update these results once we figure it out through our open-source implementation repository.