MVP-BERT: Multi-Vocab Pre-training for Chinese BERT

Wei Zhu[1] *
[1] East China Normal University, China

Abstract

Despite the development of pre-trained language models (PLMs) significantly raise the performances of various Chinese natural language processing (NLP) tasks, the vocabulary (vocab) for these Chinese PLMs remains to be the one provided by Google Chinese BERT (Devlin et al., 2019), which is based on Chinese characters (chars). Second, the masked language model pre-training is based on a single vocab, limiting its downstream task performances. In this work, we first experimentally demonstrate that building a vocab via Chinese word segmentation (CWS) guided sub-word tokenization (SGT) can improve the performances of Chinese PLMs. Then we propose two versions of multi-vocab pre-training (MVP), Hi-MVP and AL-MVP, to improve the models' expressiveness. Experiments show that: (a) MVP training strategies improve PLMs' downstream performances, especially it can improve the PLM's performances on span-level tasks; (b) our AL-MVP outperforms the recent AMBERT (Zhang & Li, 2020) after large-scale pre-training, and it is more robust against adversarial attacks.

1 Introduction

The pre-trained language models (PLMs), including BERT (Devlin et al., 2019) and its variants (Yang et al., 2019; Liu et al., 2019), have been proven beneficial for many natural language processing (NLP) tasks, such as text classification, question answering (Rajpurkar et al., 2018), natural language inference (NLI) (Bowman et al., 2015) and relation extraction (Zhu et al., 2020), on English, Chinese and many other languages. Although they bring impressive improvements for Chinese NLP tasks, most Chinese PLMs still use the vocabulary (vocab) provided by Google Chinese BERT (Devlin et al., 2019). Google Chinese BERT is a character (char) based model since it splits the Chinese characters with blank spaces. In the pre-BERT era, a part of the literature on Chinese natural language processing (NLP) first do Chinese word segmentation (CWS) to divide the text inputs into sequences of words and use a word-based vocab in NLP models (Xu et al., 2015; Zou et al., 2013). There are many arguments on which vocab a Chinese NLP model should adopt.

The advantages of char-based models are apparent. First, char-based vocab is smaller, thus reducing the model size. Second, it does not rely on CWS, thus avoiding word segmentation error, which can directly result in performance gain in span-based tasks such as named entity recognition (NER). Third, char-based models are less vulnerable to data sparsity or the presence of out-of-vocab (OOV) words and thus less prone to over-fitting (Li et al., 2019). However, word-based model has its advantages. First, it will result in shorter sequences than char-based counterparties, thus are faster. Second, words are less ambiguous, thus helping models learn the semantic meanings of words. Third, with a word-based model, exposure biases may be reduced in text generation tasks (Zhao et al., 2013). Another branch of literature tries to balance the two by combining word-based embedding with char-based embedding (Yin et al., 2016; Dong et al., 2016).

This article tries to strike a balance between the char-based and word-based models and provides alternative approaches for pre-training Chinese PLMs. We experiment on two approaches to build a vocab for Chinese PLMs: (1) following Devlin et al. (2019), separate the Chinese chars with white spaces, and then learn a sub-word tokenizer (denote as *CHAR*); (2) first segment the sentences with a CWS toolkit like jieba[1], and then learn a

*Contact: 52205901018@stu.ecnu.edu.cn.

[1]https://github.com/fxsjy/jieba

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 260–269
August 5–6, 2021.

sub-word tokenizer (denoted as *SGT*); (3) do CWS and keep the high-frequency words as tokens and low-frequency words will be tokenized by *SGT* (denoted as *SEG*). See Figure 1 for their workflow of processing an input sentence. The experiments show that *SGT* is best suited for PLMs.

Inspired by the previous work that incorporates multiple vocabularies (vocabs) or naturally combines multiple vocabs (Yin et al., 2016; Dong et al., 2016; Zhang & Li, 2020), we also investigate a series of strategies, which we will call Multi-Vocab Pre-training (MVP) strategies. The first version of MVP incorporates a hierarchical structure to combine the char-based vocab and word-based vocab. From the viewpoint of model forward pass, Chinese characters' embeddings are aggregated to form the vector representations of multi-gram words or tokens, which are fed into transformer encoders. Then the word-based vocab will be used in masked language model (MLM) training. The second version of MVP (denoted as AL-MVP) is to employ an additional vocab to form an auxiliary loss term in MLM, enhancing the PLM's ability to capture the contextual information.

Extensive experiments and ablation studies are conducted. We select BPE implemented by sentencepiece[2] as the sub-word tokenization model, and Albert (Lan et al., 2019) (tiny and base model) as our PLMs. Pre-training is done on Chinese Wikipedia corpus[3] (C-1), and a larger corpus we collect (C-2). The MVP strategies are compared on a series of Chinese benchmark datasets, two of which are sentence classification (CLS) tasks, two are named entity recognition (NER) tasks, and the remaining two are machine reading comprehension (MRC) tasks. The experimental results reveal the following take-aways: 1) combining CWS and sub-word tokenization yields the best vocab for Chinese PLMs; 3) MVP strategies can improve a single-vocab model on all three types of tasks.

We now summarize the following contributions in this work.

- We validate that combining CWS and sub-word tokenization is a better way for building vocabs for Chinese PLMs.
- We propose the novel MVP pre-training strategies for enhancing the Chinese PLMs, and they are proven to be effective.

[2]https://github.com/google/sentencepiece
[3]https://dumps.wikimedia.org/zhwiki/latest/

2 RELATED WORK

Since Devlin et al. (2019), a large amount of literature on pre-trained language models appear and push the NLP community forward with a speed that has never been witnessed before. Peters et al. (2018) is one of the earliest PLMs that learns contextualized representations of words. GPTs (Radford et al., 2018, 2019) and BERT (Devlin et al., 2019) take advantage of Transformer (Vaswani et al., 2017). GPTs are uni-directional and make predictions on the input text in an auto-regressive manner, and BERT is bi-directional and makes predictions on the whole or part of the input text. At its core, what makes BERT so powerful are the pre-training tasks, i.e., Mask language modeling (MLM) and next sentence prediction (NSP), where the former is more important than the latter. Since BERT, a series of improvements have been proposed. The first branch of literature improves the model architecture of BERT. ALBERT (Lan et al., 2019) makes BERT more light-weighted by embedding factorization and progressive cross-layer parameter sharing. Zaheer et al. (2020) improve BERT's performance on longer sequences by employing sparser attention.

The second branch of literature improves the training of BERT. Liu et al. (2019) stabilize and improve the training of BERT with a larger corpus. More work has focused on new language pre-training tasks. ALBERT (Lan et al., 2019) introduce sentence order prediction (SOP). StructBERT (Wang et al., 2019) designs two novel pre-training tasks, word structural task and sentence structural task, to learn better representations of tokens and sentences. ERNIE 2.0 (Sun et al., 2019) proposes a series of pre-training tasks and applies continual learning to incorporate these tasks. ELECTRA (Clark et al., 2020) has a GAN-style pre-training task for efficiently utilizing all tokens in pre-training. Our work is closely related to this literature branch by designing a series of novel pre-training objectives by incorporating multiple vocabularies. Our proposed method is off-the-shelf and can be easily incorporated with other pre-training tasks.

Another branch of literature looks into the role of words in pre-training. Although not mentioned in Devlin et al. (2019), the authors propose whole word masking in their open-source repository, which is effective for pre-training BERT. In SpanBERT (Joshi et al., 2019), text spans are masked

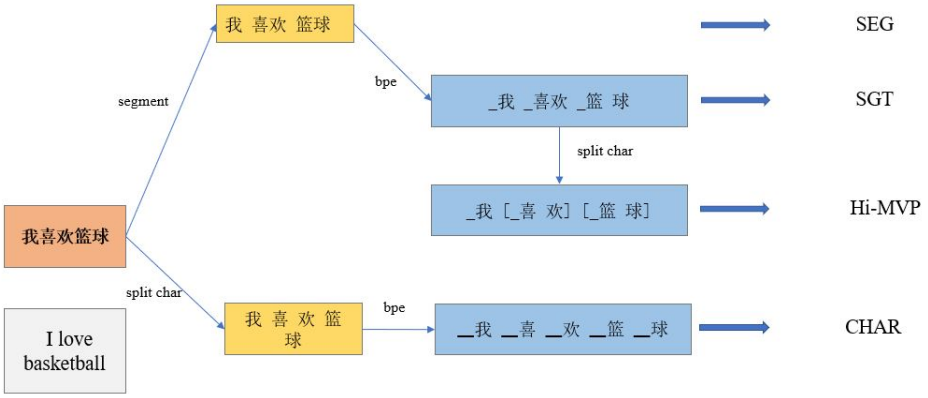


Figure 1: An illustration of how to process input sentence into tokens under different methods we define.

in pre-training, and the learned model can substantially enhance the performances of span selection tasks. It is indicated that word segmentation is vital for Chinese PLMs. Cui et al. (2019) and Sun et al. (2019) both show that masking tokens in the units of natural Chinese words instead of single Chinese characters can significantly improve Chinese PLMs. Liu et al. (2019) apply CWS to build a vocab that can improve Chinese-English translation performance. AMBERT (Zhang & Li, 2020) propose to leverage vocabs of different granularity in encoding sentences and improve the pre-training. In this work, compared to literature, our contributions are: (a) we find that CWS and sub-word tokenization can improve the pre-trained models' performances on downstream tasks. (b) we propose MVP pre-training tasks, which are proven to improve the expressiveness of pre-trained models and downstream performances.

3 Our methods

This section presents our methods for rebuilding the vocab for Chinese PLMs and introducing our series of MVP strategies.

3.1 Building the vocabs

We investigate four workflows to process the text inputs, each corresponding to a different vocab (or a group of vocabs) (Figure 1). We first introduce the single vocab models, *CHAR*, *SEG* and *SGT*.

For char-based vocab *CHAR*, Chinese characters in the corpus are treated as words in English and are separated with blank spaces, and a sub-word tokenizer is learned.[4] This method is essentially how BERT (Devlin et al., 2019) builds the Chinese vocab.

SGT (short for segmentation guided tokenization) requires the corpus sentences to be segmented with a CWS tool, and a sub-word tokenizer like BPE is learned on the segmented sentences. Some natural Chinese words in *SGT* will be split into pieces, but there are still many tokens with multiple Chinese chars.

Finally, *SEG* (short for segmentation) with size N is built with the following procedures: (a) do CWS on the corpus; (b) for long-tail Chinese words and non-Chinese tokens, tokenize them into tokens that have high frequencies; (c) sort the vocab via frequency, and if the most frequent N words or tokens can cover R percent of the corpus[5], then take them as vocab; if not, then re-do (b).

Note that *SEG* is essentially how AMBERT (Zhang & Li, 2020) builds the vocab for their Chinese PLM. However, they do not learn a sub-word tokenizer after CWS, thus making our *SGT* different from theirs. We will use experiments to show that our *SGT* yields comparably better PLMs.

3.2 Multi-vocab pre-training (MVP)

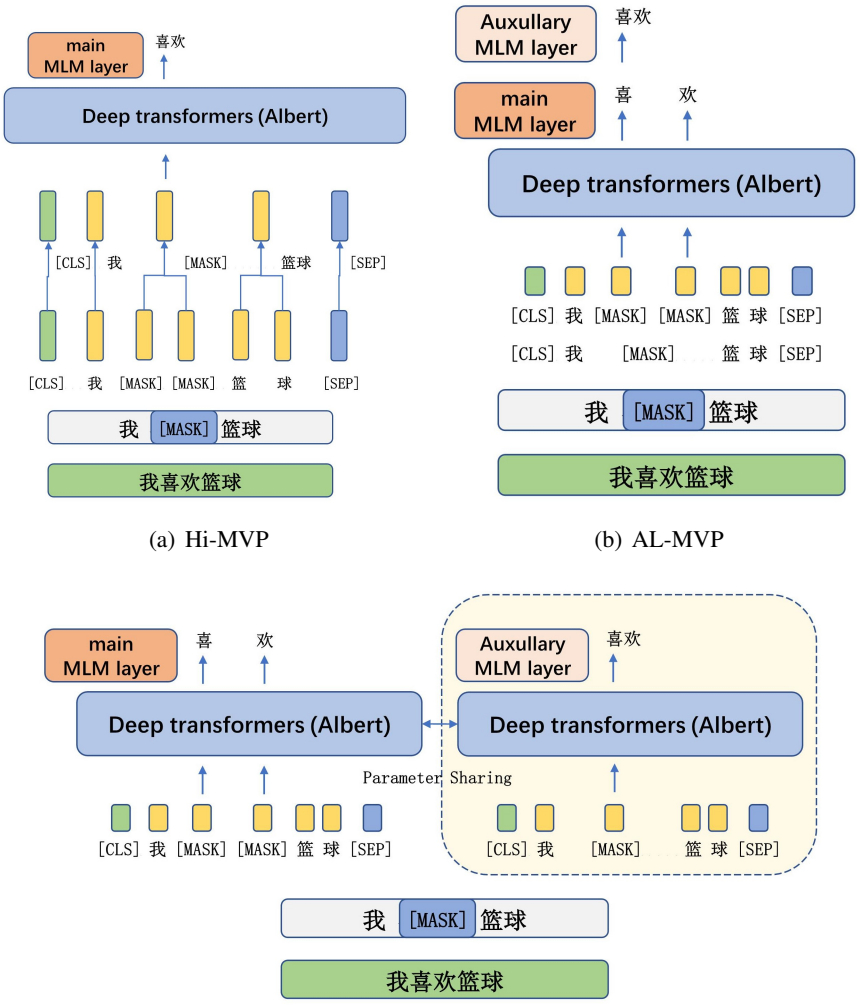
In this subsection, we will introduce MVP, a series of natural extensions to the MLM task by Devlin et al. (2019).

3.2.1 Hierarchical MVP

We first introduce hierarchical MVP (Hi-MVP). Figure 2(a) depicts the architecture of Hi-MVP, and Figure 1 depicts its procedure for processing

[4]Here the sub-word tokenizer mainly learns how to deal with non-Chinese tokens.

[5]This follows the implementation of BPE, which also asks the tokenizer to cover most of the corpus. The ratio is usually set as 99.99%.



(a) Hi-MVP (b) AL-MVP

(c) AMBERT

Figure 2: The architectures for the two versions of MVP strategies. The first two are ours, and the third one is AMBERT's.

input sentences. Two vocab, a more fine-grained vocab V_f, and a more coarse-grained vocab V_c, are combined hierarchically. Sequences are first tokenized via V_c, and then the Chinese tokens (if containing multiple Chinese chars) are split into single chars. Thus V_f consists of Chinese chars and non-Chinese tokens from V_c. Then Chinese chars and non-Chinese tokens are embedded into vectors. The representations of chars inside a token are aggregated into the representation of this token, further fed into the transformer encoder. We apply a convolution network (with kernel size 3 and #channels equally the embedding size) and max-pooling to convert the char sequence into a fixed token level representation in this work.

During MLM task, whole word masking is applied. That is, we will mask 15% of the tokens in the V_c. For example, in Figure 2(a), "_喜欢" (like) is masked, thus in the char sequence, two tokens "_喜" and "欢" are masked. A classifier is designated to predict the masked V_c token "_喜欢". Let $\mathbf{x}$ and $\mathbf{y}$ denote the sequences of tokens with lengths l_x and l_y, for the same sentence under V_c and V_f, in which a part of tokens are masked. Denote $\mathbf{x}^{mask}$ as the masked tokens under V_c. The loss function for MVP_{hier} is

$$\begin{aligned} &\min_{\theta} -\log \mathrm{P}_{\theta}(\mathbf{x}^{mask}|\mathbf{x},\mathbf{y}) \\ \approx &\min_{\theta} -\sum_{i=1}^{l_x} \mathrm{I}_i \log \mathrm{P}_{\theta}(x_i^{mask}|\mathbf{x},\mathbf{y}), \end{aligned} \tag{1}$$

in which I_i^x is a variable with binary values indicating whether the i-th token is masked in $\mathbf{x}$.

3.2.2 Auxiliary loss MVP

Figure 2(b) depicts another version of MVP. In this method, a sentence is tokenized and embedded in a fine-grained V_f (e.g., a char-based vocab), and an MLM task on V_f is conducted. However, different from the vanilla MLM, an auxiliary MLM loss objective based on a more coarse-grained vocab V_c is added. Thus, we call this method Auxiliary loss MVP (AL-MVP).

For example, encoded representations of the chars "_喜" and "欢" inside the word "_喜欢" is aggregated to the vector representation of the word, and an auxiliary MLM layer is tasked to predict the word-based on V_c. For the aggregator in the example, we adopt the BERT-style pooler, which uses the starting token's representation to represent the word's representation.[6] Denote $\mathbf{x}^{mask}$ and $\mathbf{y}^{mask}$ as the masked tokens under V_f and V_c, respectively. The loss function for MVP_{obj} is as follows:

$$\begin{aligned}
&\min_{\theta} -\log \mathrm{P}_\theta(\mathbf{x}^{mask}, \mathbf{y}^{mask}|\mathbf{x}) \\
\approx &\min_{\theta} -\sum_{i=1}^{l_x} \mathrm{I}_i^x \log \mathrm{P}_\theta(x_i^{mask}|\mathbf{x}) \\
&- \lambda * \sum_{i=1}^{l_y} \mathrm{I}_i^y \log \mathrm{P}_\theta(y_i^{mask}|\mathbf{x}), \qquad (2)
\end{aligned}$$

in which I_i^x and I_i^y are variables with binary values indicating whether the i-th token is masked in sequence x and y, respectively. Here λ is the coefficient which measures the relative importance of the auxiliary MLM task.

Note that AL-MVP is different from AMBERT's architecture (Figure 2(c)). In AMBERT, a sequence has to be encoded twice with different vocabs. Meanwhile, AL-MVP is a plug-in pre-training strategy, and during inference, the PLM is the same as the original PLM.

We will denote the model pre-trained with Hi-MVP strategy and vocab V as Hi-MVP(V) for notational convenience. AL-MVP with a fine-grained vocab V_f and a coarse-grained vocab V_c are denoted as AL-MVP(V_f, V_c).

4 Experiments

4.1 Setup

Two corpora are used for pre-training. The first one is Chinese Wikipedia (C-1). We conduct most of the experiments and ablation studies on this corpus. Finally, we will use the other corpus (C-2) to match the SOTA performances. C-2 has 25 million documents, thus it has approximately the same size as the Chinese corpus in AMBERT (Zhang & Li, 2020). [7]

CHAR's vocab size is set at 21128, which is the same with Google Chinese BERT. We consider three vocab sizes for *SGT*: {21,128, 31,692, 72,635}. We will show in experiments that *SGT* works best with vocab size 31,692. Moreover, for the experiments with AL-MVP, we will only consider *SGT* with vocab size 31,692. We set the vocab size of *SEG* to be 72,635, which is the same as AMBERT. Table 1 reports the basic statistics for the tokens in these vocabs. As the vocab size goes up, As the vocab size goes up, the vocab will include more and more phrase-level tokens (# Chinese chars ≥ 2).

For Hi-MVP, we consider Hi-MVP(*SGT*) and Hi-MVP(*SEG*). For AL-MVP, we consider AL-MVP(*CHAR*, *SGT*), AL-MVP(*CHAR*, *SEG*), and AL-MVP(*SGT*, *SEG*). The relative importance coefficient λ in Eq. 2 is tuned from the set {0.1, 0.5, 1.0, 2.0, 10.0} via training on a small corpus with 100k sentences and a small dev corpus with 5k sentences. We finally select $\lambda = 0.5$ for all models.

For pre-training, whole word masking is adopted, and a total of 15% of the words (from CWS) in the corpus are chosen. Furthermore, following BERT (Devlin et al., 2019), 80% of the chosen words are masked, a random word replaces 10%, and the rest remain unchanged. For AL-MVP, 1/3 of the time masked tokens from the fine-grained vocab are predicted, and 1/3 of the time masked tokens from the coarse-grained vocab are predicted, and for the rest of the time, masked tokens from both vocabs are predicted.

In this article, all models use the ALBERT as the encoder. We use two different settings. The first is for a smaller ALBERT model (ALBERT-tiny). The number of layers is 3, the embedding size is 128, and the hidden size is 256. We use this setting for extensive comparisons and ablation studies. Then we use the second model configuration, which is the same as ALBERT base. We pre-trained the best model from AL-MVP and show that our method also works for large language models.

[6] Due to limited resources available, we leave to future work to investigate whether alternative aggregators can bring improvements.

[7] Since AMBERT (Zhang & Li, 2020) does not open-source their corpus, we collect the corpus ourselves. C-2 consists of Chinese Wikipedia and news articles we crawled from the web.

Vocab	vocab size	zh words	zh words (len=1)	zh words (len=2)	zh words(len>=3)	other
CHAR	21,128	48.59	48.59	0	0	51.39
SGT	21,128	89.02	38.61	44.06	6.32	10.98
SGT	31,692	88.49	27.56	51.36	9.57	11.49
SGT	72,635	85.72	17.43	36.72	31.58	14.27
SEG	72,635	89.53	16.86	38.93	33.74	10.47

Table 1: The compositions of different vocabs.

Other ALBERT configurations remain the same with ALBERT (Lan et al., 2019). The pre-training hyper-parameters are almost the same with ALBERT (Lan et al., 2019) and the maximum sequence length is 512. Here, the sequence length is counted under the more fine-grained vocab for AL-MVP. The batch size is 1024, and all the models are trained for 12.5k steps. The pre-training optimizer is LAMB, and the learning rate is 1e-4. For finetuning, the sequence length is 256, the learning rate is 2e-5, the optimizer is Adam (Kingma & Ba, 2015), and the batch size is set as the power of 2 so that each epoch contains less than 500 steps. Each model is run on a given task 10 times, and the average performance scores are reported for reproducibility.

4.2 Baseline models

The first group of baselines is the original Google Chinese BERT (Devlin et al., 2019), with different vocabs. The second one is AMBERT (Zhang & Li, 2020), a pre-trained model with two vocabs of different granularity. For fair comparison, we pre-train the baselines ourselves, with the same corpus.

4.3 benchmark tasks

For downstream tasks, we select two sentence pair classification (CLS) tasks: (1) XNLI from Conneau et al. (2018) ; (2) LCQMC (Liu et al., 2018). We also investigate two named entity recognition (NER) tasks. MSRA NER (MSRA) (Levow, 2006) is from open domain, and CCKS NER[8] (CCKS) is collected from medical records. For machine reading comprehension (MRC) tasks, we consider two benchmark datasets, CMRC2018 (Cui et al., 2019) and ChID (Zheng et al., 2019).

4.4 Results for different vocabs

Table 3 report the results of pre-training ALBERT-tiny with a series of different vocabs. We can see that *SGT* obtains the best results on CLS, while *CHAR* and *SGT* have comparable results for span-level tasks NER and MRC. Even though the model with *SEG* has more parameters than *SGT*, it consistently under-performs *SGT*. The above results indicate two conclusions. First, CWS alone can not build a proper vocab for Chinese BERT. Second, sub-word tokenizers learned on the segmented Chinese corpus can decompose long-tail words into tokens while keeping meaningful phrases as it is, improving the downstream performances of ALBERT.

Also, Table 3 reports *SGT*'s performances using different vocab sizes. The results show that vocab size 31,692 is best suited for Chinese PLMs. When the *SGT*'s vocab size goes up, the less frequent tokens will not receive enough training, thus affecting the downstream performances. When the *SGT*'s vocab size goes down, it is essentially similar to *CHAR*. Thus it can not leverage phrasal information of the Chinese language. Thus, for the experiments in the rest of the paper, we only use *SGT* with vocab size 31,692.

SGT has the efficiency advantage over *CHAR*. We now make inference on the LCQMC test set using batch size 1[9], and the sequence length is kept as it is. We can observe that *SGT* has a 1.25x inference speed up than *CHAR*.

4.5 Results for MVP

In this subsection, we analyze results for our MVP strategies. We can see from Table 2 that when trained using the same corpus, our Hi-MVP's performance can match the AMBERT's performances. Note that AMBERT has twice the computational complexity of our Hi-MVP. Our Hi-MVP encoders the sentence from char level to phrase level, thus understanding the components of the sentence.

Note that Hi-MVP's pre-training works on the phrase level; thus, it does not perform well on the span level tasks. Table 3 shows that the AL-MVP strategy can generally improve all tasks' results, es-

[8]https://biendata.com/competition/CCKS2017 2/

[9]This is consistent with the online scenarios of the industry since user queries usually come one by one.

task	CLS		NER		MRC	
task name	LCQMC	XNLI	MSRA	CCKS	CMRC2018	ChID
metric	Acc.	macro F1	exact F1	exact F1	EM	Acc.
SGT (31,692)	79.79	60.19	81.07	85.74	61.64	70.97
AMBERT	80.64	60.89	81.57	86.34	62.86	72.45
Hi-MVP(*SGT*)	80.56	60.98	81.35	86.82	62.65	72.43
Hi-MVP(*SEG*)	80.35	60.57	81.48	86.56	62.48	72.31
AL-MVP(*CHAR*, *SGT*)	80.93	61.43	81.47	86.97	62.93	72.65
AL-MVP(*CHAR*, *SEG*)	81.05	61.14	81.83	86.49	**63.32**	72.87
AL-MVP(*SGT*, *SEG*)	**81.56**	**61.77**	**82.21**	**87.24**	63.29	**73.05**

Table 2: The main experimental results for our MVP strategies. Our methods outperform AMBERT, even though they require less computational resources for pre-training.

task	CLS		NER		MRC	
task name	LCQMC	XNLI	MSRA	CCKS	CMRC2018	ChID
metric	Acc.	macro F1	exact F1	exact F1	EM	Acc.
CHAR (21,128)	77.85	59.22	**81.14**	85.63	61.23	**71.05**
SGT (31,692)	**79.79**	**60.19**	81.07	**85.74**	**61.64**	70.97
SGT (21,128)	79.27	59.71	79.07	83.96	61.37	70.76
SGT (72,635)	79.04	59.45	78.79	83.41	60.89	70.51
SEG (72,635)	79.16	59.32	78.63	83.32	60.72	70.28

Table 3: Results for different vocabs, when used for ALBERT-tiny pre-training.

task name	LCQMC	XNLI	MSRA	CCKS	CMRC2018	ChID
AL-MVP(*SGT*, *SEG*)	**81.56**	**61.77**	**82.21**	**87.24**	**63.29**	**73.05**
AL-MVP(*SGT*, *SEG*)-1	79.79	60.19	81.07	85.74	61.64	70.97
AL-MVP(*SGT*, *SEG*)-2	80.78	60.86	81.49	86.23	62.08	71.63

Table 4: Ablation studies on the AL-MVP's pre-training strategies.

task name	LCQMC	XNLI	MSRA	CCKS	CMRC2018	ChID
Google BERT	86.72	77.64	93.61	90.25	70.08	82.04
RoBERT-wwm-ext	86.23	78.57	94.82	91.56	72.63	83.62
AMBERT (Zhang & Li, 2020)	-	-	-	-	73.25	86.62
AMBERT (ours)	86.95	78.93	95.39	91.74	73.08	85.31
AL-MVP(*SGT*, *SEG*)	**87.68**	**79.75**	**95.94**	**92.53**	**73.82**	**86.76**

Table 5: The performances of models with large scale pre-training.

Metric	AMBERT		AL-MVP(*SGT*, *SEG*)	
(↑ better)	LCQMC	XNLI	LCQMC	XNLI
original score	86.95	78.93	87.68	79.75
after-attack score	15.43	16.39	17.34	18.82
#queries	66	73	74	82

Table 6: Results on the adversarial robustness. "Query Number"denotes the number of queries the attack system made to the target model and a higher number indicates greater difficulty.

pecially on span-level tasks. Also, our two versions of AL-MVP models can outperform AMBERT on most of the tasks. AL-MVP asks the model to learn a more general representation that can work with different vocabs, making the model better understand a token's relation with its contexts.

Among the two AL-MVP models, AL-MVP(*SGT*, *SEG*) performs best on five of the six tasks. On CMRC2018, the performance of AL-MVP(*SGT*, *SEG*) is very close to AL-MVP(*CHAR*, *SEG*). AL-MVP(*SGT*, *SEG*) maintains the *SGT*'s advantage on CLS tasks while improving NER and MRC via AL-MVP pre-training.

4.6 Ablation on the pre-training strategies of AL-MVP

For AL-MVP, we emphasize that cross-vocab MLMs is essential for the pre-training. Thus, we compare AL-MVP(*SGT*, *SEG*) with two other versions. First, AL-MVP(*SGT*, *SEG*)-1 keeps the main MLM layer in Figure 2(b), that is, to only make MLM predictions on the more fine-grained vocab;[10] Second, AL-MVP(*SGT*, *SEG*)-2 only keep the auxiliary MLM layer in Figure 2(b), that is, to only make MLM predictions on the more coarse-grained vocab. Table 4 reports that AL-MVP(*SGT*, *SEG*) achieves the best results on all 6 tasks. The results show that MLM pre-training that combines both vocabs can effectively improve the PLM's language understanding abilities and downstream performances.

4.7 Large scale pre-training

In section, we report the pre-training results on C-2, a large-scale corpus matching the size of AMBERT's corpus. Table 5 reports the performances of ALBERT-base. We first directly report the results of AMBERT from Zhang & Li (2020) on the CMRC2018 and ChID tasks. Besides, to eliminate the factor of different training corpus, we also train AMBERT on the C-2 corpus. The results show that our AL-MVP(*SGT*, *SEG*) model outperforms both AMBERT models. Note that we only require half the GPU time for AMBERT training, and the inference speed of AL-MVP(*SGT*, *SEG*) is 2.15x of AMBERT.

4.8 Robustness over adversarial attacks

We claim that our AL-MVP training strategy can ask the ALBERT encoder to efficiently draw information from contexts into token representations, thus improving the expressiveness. Thus it is a fair reasonable that AL-MVP pre-trained models should be more robust to adversarial attacks. This subsection leverages the TextFooler framework (Jin et al., 2020) to conduct black-box attacks on the LCQMC and XNLI datasets. As shown in Table 6, we report the original performance, after-attack performance, and the number of queries needed by TextFooler to attack each model. We can see that AL-MVP(*SGT*, *SEG*) increases the number of queries needed to attack by a clear margin. Compared with AMBERT, our AL-MVP(*SGT*, *SEG*) demonstrates robustness improvements.

[10]This model is essentially the vanilla ALBERT-tiny with vocab *SGT*.

5 Conclusions

In this work, we propose a series of novel pre-training methods called MVPs, which leverage multiple vocabularies in the language model pre-training. To select the vocabs for MVP pre-training, we first conduct experiments to validate *SGT*, which combines Chinese word segmentation and sub-word tokenization, works best for the Chinese language model pre-training. We then use experiments to show that our proposed MVP methods can achieve better performances than AMBERT with less computational resources. Also, we show our MVP method can improve the pre-trained model's robustness against adversarial attacks.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *arXiv e-prints*, art. arXiv:1508.05326, August 2015.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv e-prints*, art. arXiv:2003.10555, March 2020.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL https://www.aclweb.org/anthology/D18-1269.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. Pre-Training with Whole Word Masking for Chinese

BERT. *arXiv e-prints*, art. arXiv:1906.08101, June 2019.

Yiming Cui, T. Liu, L. Xiao, Zhipeng Chen, Wentao Ma, W. Che, S. Wang, and G. Hu. A span-extraction dataset for chinese machine reading comprehension. In *EMNLP-IJCNLP*, 2019.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

C. Dong, Jiajun Zhang, C. Zong, M. Hattori, and Hui Di. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *NLPCC/ICCPOL*, 2016.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*, 2020.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICML*, 2015.

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

Gina-Anne Levow. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pp. 108–117, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W06-0115.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. Is Word Segmentation Necessary for Deep Learning of Chinese Representations? *arXiv e-prints*, art. arXiv:1905.05526, May 2019.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. LCQMC:a large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1952–1962, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1166.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, art. arXiv:1907.11692, July 2019.

Zihan Liu, Yan Xu, Genta Indra Winata, and Pascale Fung. Incorporating word and subword units in unsupervised machine translation using language model rescoring. In *WMT*, 2019.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv e-prints*, art. arXiv:1806.03822, June 2018.

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

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. *arXiv e-prints*, art. arXiv:1907.12412, July 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv e-prints*, art. arXiv:1908.04577, August 2019.

Ruifeng Xu, Tao Chen, Yunqing Xia, Qin Lu, Bin Liu, and Xuan Wang. Word embedding composition for data imbalances in sentiment and emotion classification. *Cognitive Computation*, 7, 02 2015. doi: 10.1007/s12559-015-9319-y.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv e-prints*, art. arXiv:1906.08237, June 2019.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. Multi-granularity Chinese word embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 981–986, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1100. URL https://www.aclweb.org/anthology/D16-1100.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for Longer Sequences. *arXiv e-prints*, art. arXiv:2007.14062, July 2020.

Xinsong Zhang and H. Li. Ambert: A pre-trained language model with multi-grained tokenization. *ArXiv*, abs/2008.11869, 2020.

Hai Zhao, Masao Utiyama, Eiichiro Sumita, and Bao-Liang Lu. An empirical study on word segmentation for chinese machine translation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 248–263. Springer, 2013.

Chujie Zheng, Minlie Huang, and Aixin Sun. Chid: A large-scale chinese idiom dataset for cloze test. In *ACL*, 2019.

W. Zhu, X. Wang, Xipeng Qiu, Yuan Ni, and G. Xie. Autorc: Improving bert based relation classification models via architecture search. *ArXiv*, abs/2009.10680, 2020.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1393–1398, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1141.