Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2405–2412
Marseille, 11–16 May 2020

SiBert: Enhanced Chinese Pre-trained Language Model with Sentence Insertion

Jiahao Chen[1,2], Chenjie Cao[2], Xiuyan Jiang[*1]
[1] Fudan University, No.825 Zhangheng Road, Pudong New Area, Shanghai, China
[2] Pingan Gammalab, No.1119 South Wanpin Road, Xuhui District, Shanghai, China
17212010084@fudan.edu.cn, caochenjie948@pingan.com.cn, [*]xyjiang@fudan.edu.cn

Abstract

Pre-trained models have achieved great success in learning unsupervised language representations by self-supervised tasks on large-scale corpora. Recent studies mainly focus on how to fine-tune different downstream tasks from a general pre-trained model. However, some studies show that customized self-supervised tasks for a particular type of downstream task can effectively help the pre-trained model to capture more corresponding knowledge and semantic information. Hence a new pre-training task called Sentence Insertion (SI) is proposed in this paper for Chinese query-passage pairs NLP tasks including answer span prediction, retrieval question answering and sentence level cloze test. The related experiment results indicate that the proposed SI can improve the performance of the Chinese Pre-trained models significantly. Moreover, a word segmentation method called SentencePiece is utilized to further enhance Chinese Bert performance for tasks with long texts. The complete source code is available at https://github.com/ewrfcas/SiBert_tensorflow.
Keywords: Self-supervised tasks, Bert, Pretrained model

1. Introduction

Recently pre-trained language models like Bert (Devlin et al., 2018), XLnet (Yang et al., 2019b), Elmo (Peters et al., 2018) ,GPT (Radford et al., 2018) have been demonstrated to offer substantial performance boosts for many NLP tasks such as Machine Reading Comprehension, Named Entity Recognition, and Natural Language Inference. There are usually two steps in similar models: pre-training and fine-tuning. Model parameters are trained on unlabeled data over different pre-training tasks and then applied to different labeled downstream tasks for fine-tuning. Researchers were devoted to improving fine-tuning skills to enhance performance of downstream tasks in previous months, but actually some studies (Sun et al., 2019b) (Yang et al., 2019a) (Dong et al., 2019) have shown that self-supervised tasks during pre-training have a huge impact on the performance of fine-tuning as these tasks determine the learning method and whether pre-trained models can utilize massive unlabeled data efficiently.

Bert, the most popular language pre-trained model in NLP communities, includes two pre-training tasks: Masked LM (MLM) and Next Sentence Prediction (NSP). In the MLM task, Bert randomly masks a certain percentage of tokens in the sentences and learns to predict these masked tokens. In the NSP task, Bert learns to predict whether two sentences are adjacent. In fact, the NSP task has two obvious shortcomings: (1) Its initial target is to model the relationship between two sentences, which is usually consequential, adversative or contradictory. However, due to the long length of two sentences, there are inevitably many domain-specific words. Therefore the Bert model can judge by whether two sentences belong to a same field, not complex inference in sentence level. As a result, the NSP task is more like a simple document-level task rather than a complex sentence-level task. Actually, the NSP task usually takes only 1/10 of the total training time to achieve nearly 100% accuracy. (2) The input format of the NSP task is inconsistent with some downstream tasks. For example, in Machine Reading Comprehension tasks (MRC), the input format is always a query and passage pair, and is different from the NSP task. This setup difference would produce a deviation between pre-training and fine-tuning, and leads Bert to continue to make decisions by repeated or highly matched tokens between two sentences rather than its semantic analysis and inference ability. This wrong tendency to learn some specific rules in word-level would make the model over-fitting occurs easily especially in small datasets.

Some studies have shown customized pre-training tasks for downstream tasks can effectively help a model to capture corresponding knowledge and semantic information. Hence a new pre-training task, Sentence Insertion (SI), is proposed in this paper to replace the NSP task in BERT for MRC datasets. The SI task randomly extracts a sentence from the document as the query, and predicts the location of it as the training objective. Meanwhile, for segments with no answer when it comes to long MRC inputs in downstream tasks, the query is extracted from another document 40% of the time, and the model needs to judge if an answer exists. The SI task has advantages below: (1) It models representations in sentence level, as the model has to analyze the logical relationships between sentences to acquire an exact decision, and it can avoid the wrong tendency to rely on highly matched words. (2) It is more compatible with the query-passage pairs mode, because its input format is consistent with MRC tasks. In this mode, the query in MRC would also pay more attention to the relevant parts in the passage after the SI task pre-training, even without fine-tuning. That is, the SI task strengthen a model's search ability significantly. Finally, to further enhance the model, a Chinese word segmentation method based on SentencePiece (Kudo, 2018) is used to embed long sequences short enough. For comparison, a baseline similarly to English WordPiece is made, which is tokenized by a Chinese tokenizer pkuseg (Luo et al., 2019).

The comparison between SI and NSP is made in eight different types of Chinese NLP tasks. Due to the different pre-training corpus, a Bert-NSP model is trained as another

* Corresponding author.

baseline in the same setup under our own corpus. Two Chinese pre-training works: Bert-WWM (Cui et al., 2019) and Ernie 1.0 (Sun et al., 2019a) are also used as the baselines. SI algorithm outperforms all other pre-trained models in query-passage pairs tasks, and takes a slim lead with Bert in other tasks.

The contributions of this paper are summarized as follows:

(I) A new pre-training task SI is presented to eliminate the difference between fine-tuning and pre-training in query-passage pairs NLP tasks. This model is named SiBert. It ranks 1st place on Chinese Machine Reading Comprehension 2019 leaderboard and exceeds the official Bert baseline by nearly 20%.

(II) A Chinese word segmentation method based on SentencePiece is used in this paper for tasks with long texts, which saves a lot of memory and help the model to contain more words in a segment to provide more context information for model decisions.

(III) The optimization skills of BlockSparse (Child et al., 2019), and the fast-gelu activation function are used, enabling the model to be trained on 8 16GB Tesla v100 GPU with batch size 256, length 512. The code and model will be published open source to GitHub.

2. Related Work

2.1. Self-Supervised Tasks

Due to the high expense of labeled data, unsupervised representation learning on large-scale unlabeled text corpora in self-supervising method has become popular recently. Bert is an auto-encoding pre-trained model. It uses multi-layers transformers (Vaswani et al., 2017) to learn lexical, syntactic and semantic knowledge among texts during pre-training before fine-tuning it on downstream tasks. The performance of unsupervised representations usually depends on the self-supervised tasks. Thus, it is crucial to design these pre-training tasks with more robustness. In the past few months, research on self-supervised NLP tasks is outlined as follows:

(I) To strengthen seq2seq downstream tasks like machine translation, (Song et al., 2019) present Masked Sequence to Sequence (MASS) pre-training task for encoder-decoder based language generation. Its encoder receives a sentence with randomly masked fragment as input, and its decoder tries to predict this masked fragment.

(II) To strengthen generation task, (Dong et al., 2019) use three types of language modeling objectives: unidirectional (both left-to-right and right-to-left), bidirectional, and seq2seq prediction. Specific self-attention masks were utilized to control what context the prediction conditions on.

(III) To construct more general-purpose relation extractors for information extraction. (Soares et al., 2019) design a self-supervised task named Matching the blanks. They extract sentences that contain the same entity, then probabilistically replace each entity's mention with [BLANK] symbols and model the context information.

(IV) In terms of multi-task self-supervised learning, Ernie 2.0 implements multi-task learning during pre-training. It constructs seven self-supervised NLP tasks to capture different aspects of information in the training corpus at Word-aware, structure-aware and semantic-aware levels. These tasks share the same encoding networks, and Ernie 2.0 constantly introduces a large variety of tasks to realize continual learning in more steps.

2.2. Subword Segmentation

2.2.1. English Subword Segmentation

In subword segmentation of pre-trained models, it is hard to balance the contradiction between the vocabulary size and Out Of Vocabulary (OOV) problem. Byte-Pair-Encoding (BPE) (Sennrich et al., 2015) is a subword segmentation algorithm, which includes character-level and word-level tokens representations simultaneously. BPE first splits whole sentences into individual characters. The most frequent adjacent pairs of characters are then consecutively merged until reaching a desired vocabulary size. The vocabulary is then applied by WordPiece in Bert for tokenization.

2.2.2. Chinese Subword Segmentation

WordPiece cannot be applied to Chinese directly, as there is no space between words in Chinese. Google Chinese Bert adds space around all CJK Unicode, and single Chinese characters become equivalent to English words. In this method, vocabulary is generated by a heuristic method, and all Chinese tokens in it are single Chinese characters. This method has two major drawbacks: (1) The co-occurrence of Chinese characters would make tokens pay almost all attention to the adjacent characters which can form a word with it and ignore the sentence information. (2) Each position is occupied by only one character, and important context information is often lost in long sequence inputs, as texts are divided into several segments.

Ernie 1.0 (Sun et al., 2019a) is a Chinese pre-trained model released by Baidu. To solve the first problem above, they design a knowledge masking strategy including entity-level masking and phrase-level masking. The Entity-level strategy masks entities which are usually composed of multiple words. The Phrase-level strategy masks the whole phrase which is composed of several words standing together as a conceptual unit. Researchers of Bert-WMM (Cui et al., 2019) train a new model from the Google official Bert-base model with the whole word masking strategy which is similar to phrase-level masking as a remedy for the model to know the word boundary. These masking strategies can alleviate the first problem of subword segmentation in the last paragraph partly.

3. Our Approach

3.1. Model Architecture

3.1.1. Encoder Layer

The architecture of the model is shown as Fig. 1. Concretely, given the input sample consisted of a query with one sentence $Q = \{Sentence_k\}$ and a passage with n sentences $P = \{Sentence_i\}_{i=1}^{n}$. Q is selected from P in 60% of the time and selected randomly from another passage in 40% of the time. The segment embeddings of them are $EmbeddingA$ and $EmbeddingB$ correspondingly. The [CLS] symbol is located in the first position of the whole input, and two [SEP] symbols are located in the last position

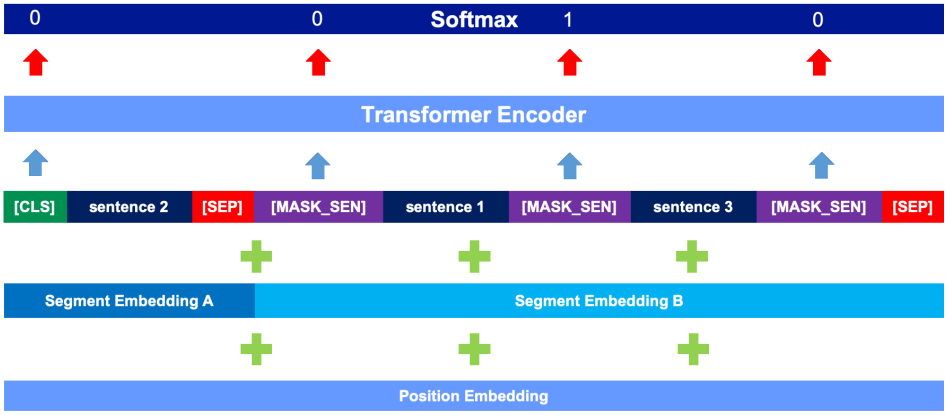


Figure 1: The detailed pre-training process of SI. The input of transformer encoder is the sum of word embedding, segment embedding and position embedding. The whole input is fed into a multi-layers transformer, and the representations of input are generated. All representations whose input is [MASK_SEN] or [CLS] have labels and only the label in the right position of them is "1".

of the query and passage part. Therefore, the whole input X can be seen as the concatenation of

$$X = [[CLS], Q, [SEP], P, [SEP]]. \quad (1)$$

The input X will go through a multi-layers transformer encoder later and outputs a sequence representation T for multi-tasks pre-training:

$$T = transformer(X), \quad (2)$$

where $X, T \in \mathbb{R}^{d_{length} \times d_{model}}$. d_{length} is the length of the sequence and d_{model} indicates the dimension of the model. BlockSparse skills (Child et al., 2019) is applied to accelerated calculation. Additionally, gelu activation function $0.5x(1 + tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$ is replaced with fast gelu $x * sigmoid(1.702x)$ in the feed-forward layers of the transformer. This replacement reduces operations of GPU and saves nearly half of the memory.

3.1.2. Task Layer

There are three pre-training tasks in total, which are Sentence Insertion (SI), Sentence-Document Relation Prediction (SDRP) and Masking Language Model (MLM). More details about MLM can be found in (Devlin et al., 2018).

3.1.3. Sentence Insertion

As mentioned above, $Q : Sentence_k$ $(60\% : k \in n, 40\% : k \notin n)$ is randomly extracted as the query input. Referring to the NSP strategy used in Bert, we extract 60% of Q from the original passage P, and 40% of Q from random sentences. Then, a [MASK_SEN] symbol is inserted to replace the original $Sentence_k$ in the passage if Q belongs to P. Later, [MASK_SEN] symbols are inserted among sentences (including the beginning and end of documents) in the passage with a 50% probability. These [MASK_SEN] symbols are used for predicting the location of the query in SI tasks. The labels of all [MASK_SEN] symbols in SI are generated during the data generation. As shown in Fig. 1, only the label in the correct position is "1", while others are all "0". Furthermore, if query Q is randomly selected from other documents, the label of the position [CLS] symbol is "1", and all other labels are "0". Whether Q belongs to this document or not, no other [MASK_SEN] symbols will be inserted into the passage, and the locations of these symbols are fixed.

Suppose there are K [MASK_SEN] and one [CLS] symbols in the input X and the output logits of SI can be written as :

$$\begin{aligned} M &= gather(T, index_{K+1}) \in \mathbb{R}^{d_{K+1} \times d_{model}}, \\ logits(M) &= softmax(W^{SI} M), \end{aligned} \quad (3)$$

where M is the output of transformer encoder which locate in the positions of [CLS] and [MASK_SEN] symbols. The cross entropy loss of SI can be computed as:

$$L = -\sum_{i \in \sigma} p(i) log(logits(M(i))), \quad (4)$$

where $p(i)$ is the real label in position i and σ is the set of pre-training data.

3.1.4. Sentence-Document Relation Prediction

It is believed that adding more tasks to pre-trained models can improve the robustness of language representation. Therefore, a specific task for [CLS] symbol is designed. This task shares the same encoder layer with SI. The output in [CLS] would go through a fully-connected layer and is used to predict the relation between the query and passage. The pairs that are labeled as "0" stand strong relevance, which means the query and passage are extracted from the same document and the query is a part of the passage. Those labeled as "1" represent weak relevance, which means the pairs are from the same document, but the query is not a part of the passage. The label "2" means that the query and passage are completely irrelevant and extracted from different documents.

3.2. SentencePiece

SentencePiece is a subword segmentation algorithm based on language model probabilities. This algorithm is used in this paper to reduce the sentence length for long text tasks.

The probability of a subword sequence S=$\{s_1, ...s_n\}$ is the product of the subword occurrence probabilities p(s_i):

$$p(S) = \prod_{i=1}^{n} p(s_i), \quad (5)$$

where all subwords s_i is from a pre-determined vocabulary and each subword occurs independently. Subword occurrence probabilities $p(s_i)$ are estimated via the EM algorithm. The most probable segmentation for the input sentence would be given by maximizing $p(S)$.

Most subwords in our pre-determined vocabulary are composed of multiple characters. An experiment was implemented before the pre-determined vocabulary size was decided. The experiment shows that as the size and average length of subwords in vocabulary increasing, the text would be embedded shorter and the performance of long text tasks would be better. The reason is that when one character occupies only one token in the previous pre-trained models, long sequence would be split into several segments. In this situation, the Bert model would lose many context information. In our pre-determined vocabulary, 2.7 characters occupy one token on average, which is a substantial improvement in context information retention.

It is noteworthy that phrase-level vocabulary cannot be applied to token-level downstream tasks directly. When the label level is tiny (for example just one Chinese character), the output of a token (usually several Chinese characters) is too large to represent such a tiny label. For this reason, SentencePiece may only be applied to sentence-level task, not token-level task at present.

4. Experiment

4.1. Pre-training Setup

4.1.1. Models

Models compared includes Google Bert, Ernie1.0, Bert-WWM, Bert + SI (SiBert), Bert-NSP, Bert + SI + SentencePiece (2SiBert), Bert + SI + SentencePiece + SDRP (3SiBert), Bert-WordPiece, and SiBert finetuned with the artificial symbol [MASK_SEN] (SiBert + AS).

Google Bert is an official Chinese pre-trained model. Ernie 1.0 and Bert-WWM are another two models released in (Cui et al., 2019) and (Sun et al., 2019a). The experiment results of Ernie1.0 and Bert-WWM are the same as the values in their papers. In SiBert, the NSP task is replaced by the SI task. For the fair comparison, another Bert baseline is trained under our own corpus and denoted as Bert-NSP. The methods and pre-training tasks listed above are added into our models in sequence to verify their effects respectively. Besides, Bert-WordPiece is another baseline in which the Pkuseg tokenizer tool is used to delimit a semantic boundary for all phrases. Spaces are set around these phrases, and then sentences are tokenized by WordPiece. In other words, these phrases are equivalent to the words in English. Finally, as many [MASK_SEN] symbols occur during pre-training, but these symbols do not exist during fine-tuning. To avoid the damage in the performance caused, these artificial symbols are inserted back between sentences in downstream tasks, to make a comparison with the original version. This model is denoted as SiBert + AS (Artificial Symbols).

4.1.2. Pre-training Details

All models have the same model size as Bert. The only change is that models are optimized with Adam optimizer using the following parameters as (Liu et al., 2019): β_1 = 0.9, β_2 = 0.98. Corpus includes several sources: 2 million news from different websites, 550k from Wikipedia, 620k from the financial field, 480k from BaiduBaike, 860k from Baike community, and 2710k from Zhihu, which is a community website similar to Quora. The vocabulary size is 20k and 70k for the character-level and phrase-level subword segmentation models respectively. Finally, models are trained on 8 16G Tesla V100 GPU for 1000k steps for 4.5 days.

4.2. Datasets and Fine-tuning Details

4.2.1. Classifications

(i) ChnSentiCorp[1]: ChnSentiCorp is a Chinese sentiment analysis dataset which aims to identify whether given sentences are positive or negative.

(ii) LCQMC (Liu et al., 2018): LCQMC is a semantic similarity task which aims to identify whether two sentences are similar semantically.

(iii) THUCNews[2]: THUCNews is a document classification dataset which is a part of THUCTC. It contains 50K news in 10 domains, including sports, finance, technology, etc.

(iv) Dbqa[3]: NLPCC-DBQA is a Question Answering (QA) task which aims to select answers for the corresponding questions. It is a query-passage pairs task with short text.

(v) XNLI (Conneau et al., 2018): XNLI is a natural language inference task to predict semantic relationship (entailment, contradiction and neutral) between two sentences.

For all classification tasks, The fine-tuning method is the same as Bert. The final hidden vector $C \in \mathbb{R}^{d_{model}}$ corresponding to the first input tokens [CLS] would be the aggregate representation and then is fed into the prediction layer. The only new parameters introduced in output layer is the weight $W \in \mathbb{R}^{K \times d_{model}}$, where K is the number of labels.

4.2.2. NER

MSRA-NER (Levow, 2006): MSRA-NER dataset is a sequence labeling task released by Microsoft Research Asia. In MSRA-NER dataset, the label includes O, B-PER, I-PER, B-ORG, I-ORG, B-LOC and I-LOC. So this NER task is treated as a 7 classification task, and we use micro average F1 as the result.

[1] https://github.com/pengming617/bert_classification

[2] http://thuctc.thunlp.org

[3] http://tcci.ccf.org.cn/conference/2016/dldoc/evagline2.pdf

Task	Metrics	SiBert		2SiBert		3SiBert		SiBert+AS		WordPiece		Bert-NSP		Bert_WWM	Ernie 1.0		Google Bert	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Dev	Test	Dev	Test
Dbqa	F1	83.2	83.8	81.4	82.6	83.1	83.2	**84.5**	**84.0**	80.4	80.8	81.4	80.3	–	82.3	82.7	80.7	80.8
Cmrc2018	EM/F1	68.3/87.6	–	–	–	–	–	**68.7/88.3**	–	–	–	65.0/85.2	–	66.3/85.6	65.1/85.1	–	65.5/84.5	–
Cmrc2019	QAC	81.2	82.4	84.9	85.7	**85.9**	**87.2**	–	–	–	–	71.9	–	–	–	–	70.6	70.0

Table 1: Results of query-passage pairs tasks. Our models (whose names include "SiBert") exceed a lot more than any other models. SiBert + AS, which contains [MASK_SEN] symbols between sentences, achieves a better result than SiBert and it proves to improve the performance of query-passage pairs tasks. 3SiBert achieves the best result in CMRC 2019, which indicates the effect of SI in MRC tasks, SentencePiece in long text tasks, and multi-tasks pre-training respectively.

Task	Metrics	SiBert		2SiBert		3SiBert		SiBert+AS		WordPiece		Bert-NSP		Bert_WWM		Ernie 1.0		Google Bert	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Chnsenticorp	Accuracy	95.5	95.9	95.5	96.1	**95.8**	**96.7**	95.1	95.6	95.1	95.2	95.1	95.2	95.1	95.4	95.2	95.4	94.6	94.3
Lcqmc	Accuracy	89.3	87.5	88.2	87.2	89.0	**88.1**	89.1	88.0	88.1	85.6	88.3	86.7	89.4	86.8	**89.7**	87.4	88.8	87.0
Xnli	Accuracy	78.7	77.9	78.7	78.2	78.6	77.8	78.5	77.7	78.2	78.0	77.8	76.8	79.0	78.0	**79.9**	**78.4**	78.1	77.2
Msra-ner	F1	**96.0**	95.2	–	–	–	–	95.6	**95.4**	–	–	95.7	95.0	–	–	95.0	93.9	94.0	92.6
Thunews	Accuracy	97.9	97.7	98.4	98.2	**98.7**	**98.5**	97.8	97.9	98.1	98.1	97.4	97.2	98.0	97.6	97.6	97.3	97.7	97.6

Table 2: Results of classification and NER tasks. Our models perform similarly as Ernie 1.0, and take a slim lead over Google Bert. Among our models, as 3SiBert utilizes all our methods, it still achieves the best values except token level tasks, which proves the effectiveness of these methods. In addition, [MASK_SEN] symbols are useless in classification and NER tasks.

Methods	Dev	Qualify	Improvement
Google Bert	70.59	70.01	–
SiBert (without WWM)	77.69	78.49	+7.10/+8.48
SiBert (with WWM)	81.17	82.44	+10.58/12.43
2SiBert	83.07	83.86	+12.48/13.85
2SiBert + DA	84.87	85.67	+14.28/15.66
3SiBert + DA	87.32	87.17	+16.73/17.16
3SiBert + DA + SSI	**90.01**	**89.71**	**+19.42/19.70**

Table 3: The ablation experiments for CMRC 2019. In this table, WWM means the Whole Word Masking for Chinese. DA indicates the data augmentation for samples with unpaired queries and passages. And SSI is the Short Sentence Insertion pretraining task mentioned in Section 4.2.3.

4.2.3. MRC

(i) CMRC 2018 (Cui et al., 2018): Chinese Machine Reading Comprehension 2018 is a span-extraction task, which is similar to SQuAD that extract a passage span for the given question.

(ii) CMRC 2019[4]: Chinese Machine Reading Comprehension 2019 is a sentence cloze-style MRC task. Given a passage and several sentences extracted from it, the model aims to complete the blanks in passage with candidate sentence. The Question Accuracy (QAC) is defined as *blanks completed exactly / total number of blanks*.

For CMRC 2018, the probabilities of each word to be the start and end of answer span are computed. The maximum sum of the two probabilities in which the end position is after the start position would be our prediction.

For CMRC 2019, we concat each candidate sentence and the passage with blanks of missing sentences as one sample. Then all blanks in the passages are replaced with [MASK_SEN] symbols, and the [CLS] symbol works as the prediction for the incompatible short sentence. Finally, we mask all other tokens to prevent the redundancy. The loss of this sentence cloze task can be written as:

$$L_{cmrc2019} = -\sum_{i=1}^{K+1} y_i \log softmax(logits_i), \quad (6)$$

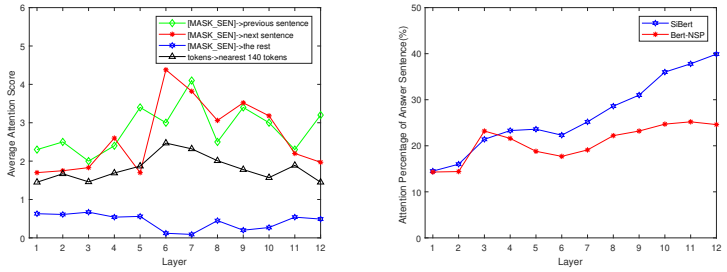
where i indicates the position of K sentences blanks [MASK_SEN] and one [CLS] symbols. For further performance improvement of CMRC 2019, we finetune the 3SiBert with another 500k steps with Short Sentence Insertion (SSI) predictions, which splits sentences with ",". Therefore, the sentence average length of SSI is reduced to 15. During the SSI fine-tuning, we masked some short sentences with [MASK_SEN] directly instead of adding [MASK_SEN] among them, which maintains consistency compared with the CMRC 2019 task. Besides, the answer is predicted dynamically, which means that the model first predicts one sentence selected from the candidates with the highest logits, and then the passage is recovered with this predicted sentence before the predication of next sentence. And this process continues until all [MASK_SEN] blanks are filled. The dynamical predicting can improve about 3% in the accuracy.

4.3. Results

For each downstream task, the best learning rate is selected from {1e-5, 3e-5, 5e-5} and the batch size is selected from {32, 64} during the fine-tuning. Each task is trained for five times with different seeds. We select the best score and the average score for dev and test result correspondingly.

Table. 1, Table. 2 shows results of query-passage pairs tasks and other tasks respectively. Details about the ablation experiments of CMRC 2019 are discussed in Table. 3. For the vacancies in Table 1 and Table 2, there are four reasons summarized as follows (1) SentencePiece cannot be applied to token level tasks. (2) [MASK_SEN] symbols have already been inserted in the CMRC 2019 task, and SiBert

[4]https://github.com/ymcui/cmrc2019



(a) (b)

Figure 2: (a) shows the average values of [MASK_SEN] symbols to tokens in different parts of passage in each layer. (b) shows the attention score percentages of query to the right answer sentence in each layer. Attention scores of special tokens like [SEP] and [CLS] are not included in the computation.

cannot predict the answers without these artificial symbols. (3) The results are not mentioned in the original papers. (4) There is no test dataset in this downstream task.

Fine-tuning skills such as dynamical predicting, ensemble learning and continual learning are limited in this section for fair comparison, so any optimized result is not included in the table. Finally, with so many methods in the experiments, it is quite hard for readers to summarize these models into a unifying conclusion. Therefore, the results are discussed in four parts.

(I) SI based models outperform all other models in query-passage pairs tasks. This is because the SI task models representations in sentence level, which is more compatible with this type of task. In other tasks, they perform similarly as Ernie 1.0.

(II) SentencePiece have huge advantages in long text datasets (CMRC 2019, THUCNews), though its performance are slightly affected in short texts. This result shows the phrase-level segmentation method in Chinese pre-trained model is able to improve tasks with long text. Besides, SentencePiece also outperforms WordPiece in all tasks, proving to be a more appropriate phrase-level Chinese word segmentation algorithm.

(III) Adding [MASK_SEN] symbols between sentences during fine-tuning can improve query-passage pairs tasks effectively, while it is useless for other tasks. These symbols can play the role of a local information observer in passage for query. More experiments will be conducted to explore these symbols in next section.

(IV) Pre-trained models would benefit tremendously from adding more tasks. A classification task SDRP is added into pre-training. Compared to the original model, 3SiBert achieves a better result in most downstream tasks.

5. Discussion

5.1. Visualization of Attention

Exploring Bert's internal mechanism (Jawahar et al., 2019) (Clark et al., 2019) by visualizing attention scores has become the popular practice recently. Hence, a series of experiments are designed to examine the effect of [MASK_SEN] symbols and the internal mechanism of SiBert .

Task	Metrics	SiBert	Bert-NSP
Original Dataset	Accuracy	87.10	80.25
Filtered Dataset	Accuracy	73.28	56.41

Table 4: Results of original and filtered datasets

5.1.1. Local Information Observer

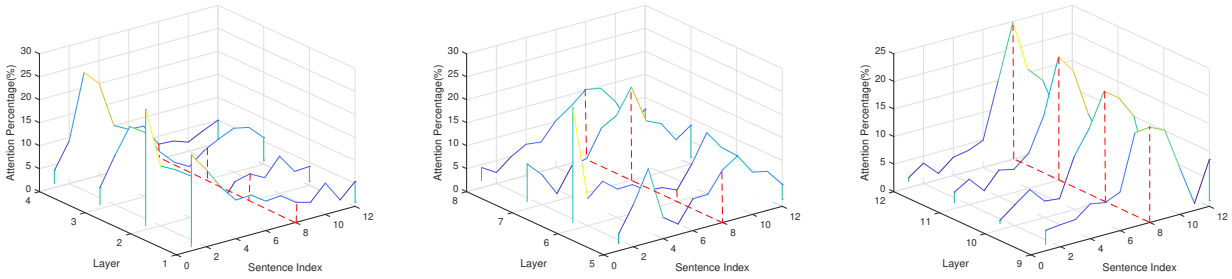
So far, there have been many studies that focus on the functions of [CLS] and [SEP] in Bert. Accordingly it is also important to clarify what [MASK_SEN] symbols attend to in SiBert by visualizing attention scores. This explains why adding this symbol into query-passage pairs tasks can improve the performance. Meanwhile, researchers can explore more reasonable fine-tuning methods base on this fact. In our experiments, attention scores are extracted from multi-layers transformer on 1k data, and then the average attention scores of [MASK_SEN] symbols to each sentence are computed. Fig. 2(a) shows the average attention scores of [MASK_SEN] symbols to the tokens in their previous sentences, next sentences and the rest of passage in each layer of the transformer. The average value of a [MASK_SEN] symbol to all passage is 1.

From Fig. 2(a), the average attention of this symbol to the previous sentence (2.96) and next sentence (2.74) is far more than the value to the rest of passage (0.61). Considering that the high value may come from the nearer position, the average attention scores of a token to its nearest 140 tokens in passage are extracted, and the result (1.8) is significantly lower than the other two values.

This experiment shows that [MASK_SEN] symbols in SiBert are equivalent to local information observers. These symbols would pay more attention to the adjacent sentences, and judge whether the sentences adjacently provide useful information. Query-passage pairs tasks would benefit from the information provided by these artificial symbols.

5.1.2. Global Information Probing

In order to understand the global attention mode of SiBert in query-passage pairs tasks, We use CMRC 2018 as the input, and compute the attention score percentage of the query to the sentence which contains real answer to measure the relevant information search ability. It should be



(a) 1-4 layers (b) 5-8 layers (c) 9-12 layers

Figure 3: These waterfall plots show a sample of Fig. 2(b), and a new dimension "Sentence Index" is added to present the distribution of the attention in each sentence. Fig. 3(a), Fig. 3(b) and Fig. 3(c) show first to fourth, fifth to eighth, and ninth to twelfth layer respectively. The colors of lines depend on its derivative, and the right sentence (8th) is marked by a red line below.

noted that this experiment is conducted before fine-tuning. The result is shown in Fig. 2(b), SiBert gives far more attention to the answer sentence than other sentences in the passage (average 15.4%). This number is also higher than Bert-NSP. Besides, this value would increase gradually as the layer goes deeper.

Fig. 3 is the visualization of an example in the last segment. This sample shows the sentence-level attention distribution in each transformer layer. In this sample, the 8th sentence which is marked by a red line below is the real answer sentence, and the 6th, 7th sentences also include the key words, as they are deceptively false answers. From Fig. 3(a), models tend to focus on the first few sentences at the beginning, and it may come from the nearer position. Then SiBert would attend to sentences with key words in middle layers in Fig. 3(b), and SiBert might be performing matches in the token level. In this example, SiBert attends closely to the answer sentence in the last four layers as shown in Fig. 3(c). This is believed to be a sign that SiBert has found the right answer by multi-turns inference. Most query-passage pairs downstream tasks would benefit from this attribute.

From these phenomena, it is obvious that since SiBert has performed many sentence-level interactive matches between the query part and the passage part during pre-training, the query can spontaneously focus on the most relevant parts in the passage even without fine-tuning. These relevant parts may contain the answer directly or provide useful information in the search of the answer, so it appears as a global information probing ability. This ability reduces the randomness in the early stage of gradient descent, and prevents the model from falling into over-fitting caused by making decisions merely by highly matched tokens.

5.2. Discussion of Tasks with Sentence Clues

Many pre-trained language models make decisions not by complex inference process, but by co-occurrence words clues. These statistical clues mainly stem from the unbalanced word distribution, and result in that some specific words are prone to a specific label erroneously. Although this method can achieve high performance in downstream tasks, it is not convincing enough. So, a new dataset is constructed in this paper to exclude the false co-occurrence clues. This dataset is the same to our pre-training data, and some heuristic methods are used to search the synonyms and concurrent words between the query and the corresponding answer sentences. Later, these examples are filtered. The dataset is split into training and test sets after the search step to ensure two datasets are in the same distribution.

The results are shown in Table 4. The dataset before and after the filtered process are denoted as Original Dataset and Filtered Dataset respectively. In the Original Dataset, the two results are very close, and the gap is only 6.85%. However in the Filtered Dataset, the gap rises sharply to 16.87%. This experiment proves that SiBert models sentence-level relationships between the query and passage. Although many samples based on word-level information are filtered, SiBert still maintains a high accuracy, while Bert-NSP shows a huge performance decline.

6. Further Conclusions

In this paper, a new NLP self-supervised task SI is proposed, and its state-of-the-art performance in query-passage pairs downstream tasks is verified by experiments. Furthermore, advantages of phrase-level pre-trained models in long text datasets are shown. Based on our work, there are also two further conclusions for future work: (1) Self-supervised tasks in different levels would help pre-trained models to avoid over-fitting caused by simply clues. (2) Downstream tasks benefit tremendously from eliminating the difference between pre-training and fine-tuning.

7. Bibliographical References

Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. (2018). Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Cui, Y., Liu, T., Xiao, L., Chen, Z., Ma, W., Che, W., Wang, S., and Hu, G. (2018). A span-extraction dataset for chinese machine reading comprehension. *arXiv preprint arXiv:1810.07366*.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., and Hu, G. (2019). Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Jawahar, G., Sagot, B., Seddah, D., Unicomb, S., Iñiguez, G., Karsai, M., Léo, Y., Karsai, M., Sarraute, C., Fleury, É., et al. (2019). What does bert learn about the structure of language? In *57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy*.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Levow, G.-A. (2006). The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117.

Liu, X., Chen, Q., Deng, C., Zeng, H., Chen, J., Li, D., and Tang, B. (2018). Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, R., Xu, J., Zhang, Y., Ren, X., and Sun, X. (2019). Pkuseg: A toolkit for multi-domain chinese word segmentation. *arXiv preprint arXiv:1906.11455*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. (2019a). Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H. (2019b). Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yang, A., Wang, Q., Liu, J., Liu, K., Lyu, Y., Wu, H., She, Q., and Li, S. (2019a). Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2346–2357.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.