NSP-BERT: A Prompt-based Few-Shot Learner Through an Original Pre-training Task —— Next Sentence Prediction

Yi Sun[*], **Yu Zheng**[*], **Chao Hao**, **Hangping Qiu**[†]
Army Engineering University of PLA, Nanjing, China
sunyi_lgdx@sina.com, zhengyu87@outlook.com
haochaoleo@163.com, qiuhp_zy@163.com

Abstract

Using prompts to utilize language models to perform various downstream tasks, also known as prompt-based learning or prompt-learning, has lately gained significant success in comparison to the pre-train and fine-tune paradigm. Nonetheless, virtually most prompt-based methods are token-level such as PET based on mask language model (MLM). In this paper, we attempt to accomplish several NLP tasks in the zero-shot and few-shot scenarios using a BERT original pre-training task abandoned by RoBERTa and other models——Next Sentence Prediction (NSP). Unlike token-level techniques, our sentence-level prompt-based method **NSP-BERT** does not need to fix the length of the prompt or the position to be predicted, allowing it to handle tasks such as entity linking with ease. NSP-BERT can be applied to a variety of tasks based on its properties. We present an NSP-tuning approach with binary cross-entropy loss for single-sentence classification tasks that is competitive compared to PET and EFL. By continuing to train BERT on RoBERTa's corpus, the model's performance improved significantly, which indicates that the pre-training corpus is another important determinant of few-shot besides model size and prompt method.[1]

1 Introduction

GPT-2 (up to 1.5B (Radford et al., 2019)) and GPT-3 (up to 175B (Brown et al., 2020)) are ultra-large-scale language models with billions of parameters that have recently demonstrated outstanding performance in various NLP tasks. Compared with previous state-of-the-art fine-tuning methods, they can achieve competitive results without any or with just a limited quantity of training data. Although studies have shown that scaling up the model improves task-agnostic and few-shot performance, some studies have shown that by constructing appropriate prompts for the model, models like BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019) can achieve similar performance despite having a parameter count that is several orders of magnitude smaller (Schick and Schütze, 2021b,a; Wang et al., 2021). Since then, the area of natural language processing has seen a fresh wave of developments, including the introduction of a new paradigm known as **prompt-based learning** or **prompt-learning**, which follows the *"pre-train, prompt, and predict"* (Liu et al., 2021) process. In zero-shot and few-shot learning, prompt-learning has achieved a lot of success. Not only does it achieve outstanding performance, prompt-learning better integrates pre-training and downstream tasks and brings NLP tasks closer to human logic and habits.

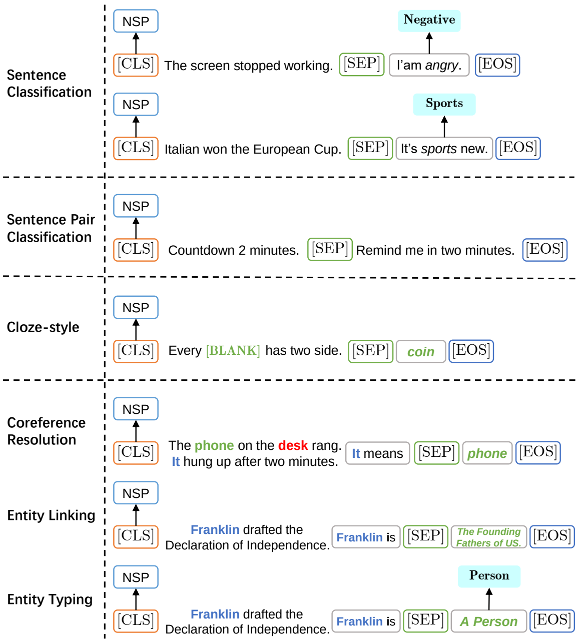
Sentence Classification
NSP
Negative
[CLS] The screen stopped working. [SEP] I'm angry. [EOS]
NSP
Sports
[CLS] Italian won the European Cup. [SEP] It's sports new. [EOS]
Sentence Pair Classification
NSP
[CLS] Countdown 2 minutes. [SEP] Remind me in two minutes. [EOS]
Cloze-style
NSP
[CLS] Every [BLANK] has two side. [SEP] coin [EOS]
Coreference Resolution
NSP
[CLS] The phone on the desk rang. It hung up after two minutes. It means [SEP] phone [EOS]
Entity Linking
NSP
[CLS] Franklin drafted the Declaration of Independence. Franklin is [SEP] The Founding Fathers of US. [EOS]
Entity Typing
NSP
Person
[CLS] Franklin drafted the Declaration of Independence. Franklin is [SEP] A Person [EOS]

Figure 1: Prompts for various NLP tasks of NSP-BERT.

The input text for the classification task, for ex-

[*] Equal contribution

[†] Corresponding author

[1] Our code and pre-trained models are publicly at: https://github.com/sunyilgdx/Prompts4Keras.

Proceedings of the 29th International Conference on Computational Linguistics, pages 3233–3250
October 12–17, 2022.

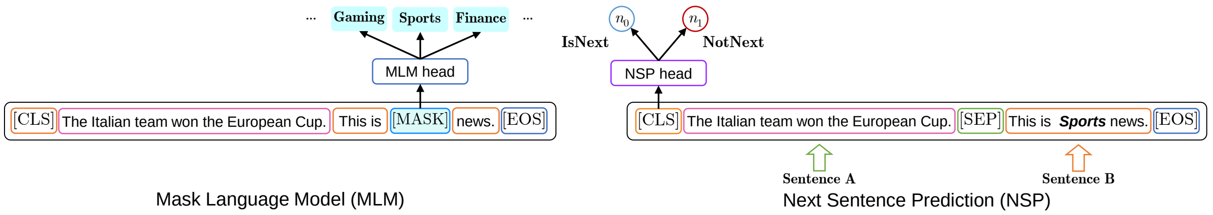
... Gaming Sports Finance ...
MLM head
[CLS] The Italian team won the European Cup. This is [MASK] news. [EOS]
Mask Language Model (MLM)
n_0 n_1
IsNext
NotNext
NSP head
[CLS] The Italian team won the European Cup. [SEP] This is Sports news. [EOS]
Sentence A
Sentence B
Next Sentence Prediction (NSP)

Figure 2: (Left) MLM task for token-level prompt-learning. (Right) NSP task for sentence-level prompt-learning.

ample, "*The Italian team won the European Cup.*", should be assigned to one of the candidate labels, such as *Gaming*, *Sports*, or *Finance*. At this point, the template "*This is* [MASK] *news.*" will be added to the original text, and the model will be asked to predict the missing word or span. The model's output will then be mapped to the candidate labels. We could utilize the pre-training tasks of several types of language models (LM) to predict the abovementioned templates, including but not limited to Left-to-right LM (GPT series (Radford et al., 2018, 2019; Brown et al., 2020)), Masked LM (BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019)), prefix LM (UniLM (Dong et al., 2019; Bao et al., 2020)) and Encoder-decoder LM (T5 (Raffel et al., 2019), BART (Lewis et al., 2020)).

Although most research on prompt-learning has been conducted, the majority of the pre-training tasks used in prompt-learning are token-level, requiring the labels to be mapped to a fixed-length token span (Schick and Schütze, 2021b,a; Cui et al., 2021). On the one hand, when the number of labels grows rapidly, this necessitates a lot of human labor. On the other hand, tasks with variable-length options make Left-to-right LM (L2R LM) or masked LM (MLM) difficult to cope with. The length of each candidate entity's description, for example, varies significantly in the entity linking task.

At the same time, we observed that there is an original sentence-level pre-training object in vanilla BERT——**NSP** (**N**ext **S**entence **P**rediction), which is a binary classification task that predicts whether two sentences appear consecutively within a document or not. Many models, like RoBERTa (Liu et al., 2019) and many others (Conneau and Lample, 2019; Yang et al., 2019; Joshi et al., 2020), have questioned and abandoned this task during pre-training. Nevertheless, based on the task's features and object, we believe it is appropriate to use in prompt-learning.

Unlike most prior works, we present NSP-BERT, a sentence-level prompt-learning method. The paper's main contributions can be summarized as follows:

- We propose the use of NSP, a sentence-level pre-training task for prompt-learning, which can ignore the uncertain length of the label words. Our NSP-BERT has a strong zero-shot learning capacity and can be applied to a wide range of tasks, which is extremely motivating for future work.
- We present NSP-tuning for single-sentence classification tasks. Without abandoning the original NSP head, binary cross-entropy loss is utilized to make the zero-shot capacity of NSP-BERT continue to few-shot by building coupled positive and negative instances.
- By using RoBERTa's corpus to continue pre-training the BERT model, although the computational cost is only about 2% of RoBERTa, our BERT$_{\mathcal{C}_{\text{B+Mix5}}}$ has been greatly improved in both zero-shot and few-shot scenarios. We believe that the effect of pre-training corpus on few-shot learning is decisive, so we suggest that all few-shot learning baselines, even if cannot use the same pre-trained model, should be based on the same pre-training corpus. In this way, a fair comparison can be made.

2 Related Work

2.1 Token-Level and Sentence-Level

Token-Level Prompt-Learning Token-level pre-training tasks, such as MLM (Shown in the left part of Figure 2) (Jiang et al., 2020; Schick and Schütze, 2021b,a) or L2R LM(Radford et al., 2019; Brown et al., 2020; Cui et al., 2021), are commonly used in token-level prompt-learning approaches. Although the expected answer may be in the form of tokens, spans, or sentences in token-level prompt-learning, the predicted answer is always generated token by token. Tokens are usually mapped to the whole vocabulary or a set of candidate words (Petroni et al., 2019; Cui et al., 2021; Han et al., 2021; Adolphs et al., 2021; Hu et al., 2021). Take PET model (Schick and Schütze, 2021b,a) as an example, the

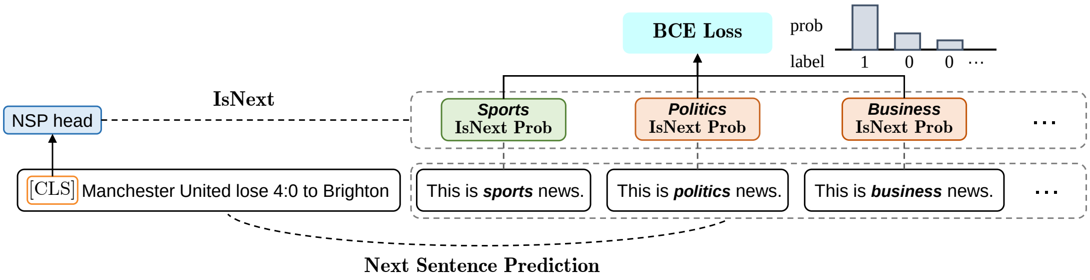

Figure 3: NSP-tuning for single-sentence classification. "*Manchester United lose 4:0 to Brighton*" is the original input, the gold label is *Sports*. The negative instances are building with wrong label *Politics*, *Bussiness*, etc.

sentiment classification input/label pair is reformulated to "$\mathbf{x}$: `[CLS]` *The Italian team won the European Cup. This is* `[MASK]` *news.* `[EOS]`, y: *Sports*".

Sentence-Level Prompt-Learning Sentence-level methods concentrate on the relationship between sentences, with the model's output usually mapped to a relationship space. As far as we know, EFL (Wang et al., 2021) is the only sentence-level model. It reformulates NLP tasks into sentence entailment-style tasks. For example, the sentiment classification input/label pair is reformulated to "$\mathbf{x}$: `[CLS]` *The Italian team won the European Cup.* `[SEP]` *This is Sports news.* `[EOS]`, y: `Entail`". The output of model is `Entail` or `Not Entail`. The EFL model can perform well on few-shot learning but relies on labeled natural language inference (NLI) datasets like MNLI (Williams et al., 2018).

2.2 Optimization methods

Automated Prompt Manually designed prompts are highly unstable. Sometimes it is necessary to be familiar with the particular task and language model in order to construct a high-quality prompt. As a result, several studies attempt to automatically search for and generate prompts. LM-BFF (Gao et al., 2021) model use conditional likelihood to automatically select labels words, and use T5 (Raffel et al., 2019) to generate templates. AUTOPROMPT (Shin et al., 2020) uses a gradient-guided search to create prompts. Compared to the discrete prompt search methods mentioned above, P-tuning (Liu et al., 2021) employs trainable continuous prompt embeddings on GPT.

Training Strategy There are many optimization methods in prompt-learning. ADAPET (Tam et al., 2021) uses more supervision by decoupling the losses for the label tokens and a label-conditioned MLM objective over the full original input. PTR (Han et al., 2021) incorporates logic rules to compose task-specific prompts with several simple sub-prompts. (Zhao et al., 2021) use content-free inputs (e.g. "N/A") to calibrate the model's output probabilities and improved the performance of GPT-2 and GPT-3.

3 Framework of NSP-BERT

Problem of MLM: Span Prediction As the most important pre-training task of BERT-like models, MLM has been used for prompt-learning in most previous studies, and achieved satisfactory results on GLUE (Wang et al., 2019) and other English datasets or benchmarks. In those English tasks, we can use just one token to map each label. But in some cases, we need more than one token.

$$\mathbf{x}_{input} = \texttt{[CLS]}\ \mathbf{x}\ \text{It was}\ \texttt{[MASK].[EOS]}$$

$$\mathbf{x}_{input} = \texttt{[CLS]}\ \mathbf{x}\ \text{这是}\ \texttt{[MASK][MASK]}\text{新闻}\texttt{.[EOS]}$$

As shown in the above example, in the first English sample, $\mathbf{x}$ is the original sentence, we can use just one `[MASK]` token to predict the label word "Sports" in a classification task. But in the second Chinese sample, we need `[MASK][MASK]` to map the label word "体育" (which has the same meaning with "Sports"), and use their probability product to represent the probability of the label (detailed description is in the Appendix A.1). As the number of `[MASK]` increases, it becomes difficult for the MLM to predict correctly. At the same time, it is impossible to compare the probability of label mapping words (spans or sentences) with different number of `[MASK]` tokens, entity linking is one of the scenarios. Therefore, especially in the Chinese task, there is a obvious gap between the pre-training and the downstream task.

3.1 Next Sentence Prediction

The next sentence prediction is one of the two basic pre-training tasks (the other is MLM) of the vanilla BERT model (Devlin et al., 2018) (Shown in the right part of Figure 2). This task inputs two sentences A and B into BERT at the same time to predict whether sentence B comes after sentence A in the same document. During specific training, for 50% of the time, B is the actual next sentence that follows A (IsNext), and for the other 50% of the time, we use a random sentence from the corpus (NotNext).

$$\mathbf{x}_{input} = \texttt{[CLS]}\mathbf{x}_i^{(1)}\texttt{[SEP]}\mathbf{x}_i^{(2)}.\texttt{[EOS]}$$

Let $\mathcal{M}$ denote the model trained on a large-scale corpus. This model is trained on both MLM task and NSP task at the same time. $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ denote sentence A and sentence B, respectively. The model's input is $\mathbf{x}_{input}$, and $q_{\mathcal{M}}$ denotes the output probability of model's NSP head. $\mathbf{s} = \mathbf{W}_{\text{nsp}}(\tanh(\mathbf{W}\mathbf{h}_{\text{[CLS]}} + \mathbf{b}))$ [2], where $\mathbf{h}_{\text{[CLS]}}$ is the hidden vector of [CLS] and $\mathbf{W}_{\text{nsp}}$ is a matrix learned by NSP task, $\mathbf{W}_{\text{nsp}} \in \mathbb{R}^{2\times H}$. The loss function of NSP task $\mathcal{L}_{\text{NSP}} = -\log q_{\mathcal{M}}(n|\mathbf{x})$, where $n \in \{\texttt{IsNext}, \texttt{NotNext}\}$.

$$q_{\mathcal{M}}(n_k|\mathbf{x}_i) = \frac{\exp s(n_k|\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})}{\sum\limits_{n} \exp s(n|\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})} \qquad (1)$$

3.2 Prompts in NSP-BERT

NSP-BERT, like other prompt-based learning methods, requires the construction of appropriate templates for various tasks. In order to make the model have better zero-shot performance and better few-shot initialization, the template's building form must closely match the original NSP task. In this section, we'll show how to construct templates for different tasks (also shown in Figure 1).

In order to apply NSP to zero or few-shot learning, we treat most tasks as multiple-choice tasks. Same as the right side in Figure 2, an NSP-BERT's input can be expressed as:

$$\mathbf{x}_{input} = \texttt{[CLS]}\mathbf{x}_i\texttt{[SEP]}p_i^{(j)}\texttt{[EOS]}.$$

We define the template $\mathcal{T}$ as a combination of input $\mathbf{x}_i$ and the prompts, $\mathcal{T}(\mathbf{x}) = \texttt{[CLS]}\mathbf{x}\ \texttt{[SEP]}$ *This is ... news.*[EOS]. Unlike prompt-tuning based on MLM (Schick and Schütze, 2021a; Gao et al., 2021) which requires mapping labels to vocabularies, for our NPS-BERT, labels can be mapped to words or phrases of arbitrary length in "...". To map labels to the prompts, we define a verbalizer as a mapping $f : \mathcal{Y} \mapsto \mathcal{P}$. The label $y_i^{(j)}$ can be mapped to prompt $p_i^{(j)} \in \mathcal{P}$.

In single-sentence classification tasks, all samples share the same label space $\mathcal{Y}$, where $|\mathcal{Y}|$ is the number of classes. For label of the jth class $y^{(j)} \in \mathcal{Y}$ can be mapped to prompt $p^{(j)}$. For those tasks where each sample corresponds to different labels, such as cloze-style task, word sense disambiguation, entity linking, we define the label space corresponding to the ith sample as $\mathcal{Y}_i$, and $y_i^{(j)} \in \mathcal{Y}_i$.

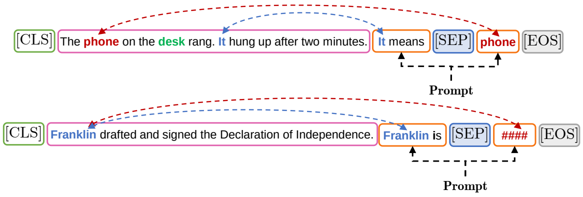
[CLS]
The phone on the desk rang. It hung up after two minutes.
It means
[SEP]
phone
[EOS]
Prompt
[CLS]
Franklin drafted and signed the Declaration of Independence.
Franklin is
[SEP]
####
[EOS]
Prompt

Figure 4: Two-stage prompt, examples in coreference resolution and entity linking/typing tasks.

In tasks such as entity linking, there are more than one entity in the sentence, in order to identify target entity words, we recommend using **two-stage prompt** (as shown in Figure 4) to indicate the target word using natural language descriptions:

- **Stage 1**: Prompt the target word at the end of sentence A. This stage's purpose is to provide enough context for the target word.
- **Stage 2**: Prompt the description of the candidate word sense in sentence B.

Let $p_{i,1}^{(j)}$ and $p_{i,2}^{(j)}$ denote the first and the second part of the prompt. The model's input is:

$$\mathbf{x}_{input} = \texttt{[CLS]}\mathbf{x}_i, p_{i,1}^{(j)}\texttt{[SEP]}p_{i,2}^{(j)}\texttt{[EOS]}.$$

For sentence-pair tasks such as text entailment and text matching, since the NSP task is in the form of sentence pairs we still use the same input as the original NSP task.

3.3 Answer Mapping

Because not all datasets can provide contrastive candidate answers (sentiments, topics, idioms, or entities), we propose two answer mapping methods, **candidates-contrast** answer mapping and **samples-contrast** answer mapping, for different situations.

[2] Devlin et al.(2018) use an additional nonlinear layer to pool the hidden vector of [CLS] for NSP task, but not mentioned in their paper.

Candidates-Contrast For datasets with multiple candidates, such as candidate sentiments, candidate topics, candidate idioms and candidate entities. For the above datasets, there is a template $p_i^{(j)}$ (or p_i) corresponding to the label $y_i^{(j)}$ (or y_i), we choose the `IsNext` probability as the output of each candidate answer. The logit of label $y_i^{(j)}$ (the value ranges from 0 to 1, but is not an actual probability) is:

$$q(y_i^{(j)}|\mathbf{x}_i) \propto q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p_i^{(j)}) \tag{2}$$

In the prediction stage, we take the highest probability output by $\mathcal{M}$ among the candidates as the final output answer where the condition is `IsNext`:

$$\begin{aligned}\hat{y}_i &= \arg\max_j \, q(y_i^{(j)}|\mathbf{x}_i) \\ &= \arg\max_j \, q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p_i^{(j)})\end{aligned} \tag{3}$$

Samples-Contrast For sentence-pair tasks, the `IsNext` output probabilities of most samples are close to 1 (see details in Appendix B.2), which makes it difficult to judge the relationship between two sentences through a single sample. So we propose the samples-contrast answer mapping method (Figure 3), to determine the label of a individual sample by contrast the probability of `IsNext` between samples. To put it simply, by **rank**ing[3] in ascending order, the samples with a relatively higher `IsNext` probability are **divide**d[4] into labels with a higher degree of matching, such as `Entailment`. On the contrary, samples with lower `IsNext` probability will be divided to labels such as `NotEntailment`. This procedure is summarized in Algorithm 1[5].

Considering the fairness of the comparative experiment, we consider two preconditions. One is that a complete development set and a test set can be obtained at the same time; the other is that only the development set can be obtained, and the test samples must be predicted one by one or batch by batch during testing. In our experiment, we use the development set to determine the thresholds of probability, and use these thresholds to predict the test set.

Algorithm 1 Samples-Contrast Answer Mapping

Input: Test set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$, Oder $o \in$ {"ascending", "descending"}, distribution of labels d, batch size bs.

Output: $\{\mathbf{x}_i, \hat{y}_i\}_{i=1}^N$

```
for $i = 1, ..., N$ do
    $q_i \leftarrow q_{\mathcal{M}}(n = \text{IsNext}|\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$
end for
$\{\mathcal{B}_j\}_{j=1}^{\lceil \frac{N}{bs} \rceil} \leftarrow$ divide $(\mathcal{D}, bs)$
for $j = 1, ..., \lceil \frac{N}{bs} \rceil$ do
    $\mathcal{B}'_j = \{\mathbf{x}_{r(1)}, ..., \mathbf{x}_{r(bs)}\} \leftarrow$ rank$(\mathcal{B}_j, q_i, o)$
    $\{B_m\}_{m=1}^M \leftarrow$ divide $(\mathcal{B}'_j, d)$
    for $i = 1, ..., bs$ do
        $\hat{y}_i \leftarrow m$ where $\mathbf{x}_i \in B_m$
    end for
end for
```

3.4 NSP-tuning

Since we treat tasks with candidates as multiple-choice problems, when we need to perform few-shot learning, we need to choose some methods to continue the initialization advantages of NSP-BERT in zero-shot. We name this method NSP-tuning used on few-shot single-sentence classification tasks, as shown in Figure 3.

Building Instances Taking the single-sentence classification as an example, for the ith sample, we take it's gold label y_i^+ as a positive instance $(\mathcal{T}(\mathbf{x}_i, y_i^+), 1)$, while taking the rest of the labels in $\mathcal{Y}$ as negative instances $\{(\mathcal{T}(\mathbf{x}_i, y_i^-), 0)\}_{y_i^- \neq y_i^+, y_i^- \in \mathcal{Y}}^{|\mathcal{Y}|-1}$ and $\{0, 1\}$ represent the labels of the binary classification. Both the positive instance and negative instances of the same sample, a total of $|\mathcal{Y}|$, will be coupled and input to the model in a same batch.

Loss function Since the output probability of `IsNext` has been already normalized to $[0, 1]$ by softmax after a nonlinear layer during pre-training, if we want to do NSP-tuning without changing the structure of the pre-training model, we need to choose the **binary cross-entropy loss** as the loss function. Of course, we can re-initialize the output of $\mathcal{M}$ to implement a multiple-choice method with linear layer+softmax cross-entropy loss same as (Radford and Narasimhan, 2018), but we think this is not conducive to preserving the zero-shot advantage of NSP to few-shot.

[3] Sort samples in ascending or descending order according to `IsNext` probability.

[4] Divide the dataset (or sample batch) into subsets according to the proportion of each label in development set.

[5] This method is currently only suitable for sentence-pair tasks, and can only be applied in zero-shot scenarios.

		English Tasks							Chinese Tasks			
		SST-2	**MR**	**CR**	**MPQA**	**Subj**	**Yahoo!**	**AGNews**	**EPR.**	**TNEWS(K)**	**CSLDCP**	**IFLY.**
Full	Majority	*50.9*	*50.0*	*50.0*	*50.0*	*50.0*	*10.0*	*25.0*	*50.0*	*6.7*	*1.5*	*0.8*
	Fine-Tuning	*93.6*	*89.0*	*89.3*	*89.3*	*97.0*	*76.5*	*94.7*	*90.0*†	*71.0*†	*68.0*†	*66.0*†
Zero	PET	67.6	65.3	**61.2**	**63.9**	**61.0**	25.6	54.5	60.7	28.0 / 35.6	22.4	34.8
	NSP-BERT	**75.6**	**74.4**	59.4	59.9	53.9	**47.0**	**77.5**	**86.9**	**51.9** / **57.0**	**47.6**	**41.6**
Few	Fine-tuning	77.9±5.9	68.0±9.4	79.1±8.9	65.2±6.3	**89.7±1.1**	61.8±1.5	82.4±1.2	78.7±5.8	51.1±1.1 / 58.0±1.4	51.7±2.1	45.1±2.2
	PET	86.0±1.6	80.0±1.6	**88.9±0.6**	83.3±2.4	86.2±1.5	64.3±1.3	84.2±0.8	82.5±2.0	54.7±1.1 / 61.2±0.9	52.6±1.2	45.9±2.1
	EFL w/ PT	**86.9±1.8**	**80.6±1.2**	88.1±0.9	**86.1±0.7**	86.0±3.3	63.0±1.2	83.8±1.3	84.8±1.6	53.2±1.5 / 59.2±1.6	52.0±1.6	47.9±1.5
	EFL w/o PT	81.2±5.1	76.1±9.1	79.2±4.0	79.1±1.6	75.1±9.4	60.8±4.2	84.6±0.7	84.6±2.1	54.7±1.3 / 60.3±1.7	53.8±0.9	49.5±1.2
	NSP-BERT	86.8±1.3	80.5±1.5	86.0±2.2	83.9±1.1	86.4±1.8	**64.5±0.5**	**85.9±0.8**	**87.7±0.7**	**55.7±1.0** / **61.6±0.9**	**55.0±1.5**	**49.5±1.1**

Table 1: Main zero-shot and few-shot learning results on single-sentence classification tasks. In addition to the accuracy, we also report the standard deviation for few-shot learning. For English tasks, we use vanilla BERT-LARGE. For Chinese tasks, we use UER's Chinese BERT-BASE. Full: full training; Zero: zero-shot; Few: few-shot; †: human performance; Majority: majority class; EFL w/ PT: few-shot tuning of EFL with pre-training on MNLI; EFL w/o PT: few-shot tuning of without pre-training on MNLI; TNEWS(K): use the keyword (K) field or not.

4 Experiment

4.1 Tasks and Datasets

English Datasets For English tasks, following (Gao et al., 2021; Hu et al., 2021; Liang et al., 2022), we choose 7 single-sentence and 5 sentence-pair English tasks. See details in Appendix B.1.

Chinese Datasets For Chinese tasks, we choose FewCLUE (Xu et al., 2021), a Chinese Few-shot Learning Evaluation Benchmark, which contains 9 NLU tasks in Chinese, with 4 single-sentence tasks, 3 sentence-pair tasks and 2 reading comprehension tasks. Additionally, we select the entity linking dataset DuEL2.0[6] to verify the word sense disambiguation ability. And we divide DuEL2.0 into two parts: DuEL2.0-L (entity linking) and DuEL2.0-T (entity typing).

4.2 Baselines

Fine-Tuning Standard fine-tuning of the pre-trained language model on the FewCLUE training set. The models are fine-tuned with cross entropy loss and using the BERT-style model's hidden vector of `[CLS]` $\mathbf{h}_{[\text{CLS}]}$ with a classification layer $\text{softmax}(\mathbf{W}\mathbf{h}_{[\text{CLS}]})$, where $\mathbf{W} \in \mathbb{R}^{|\mathcal{Y}| \times H}$, $|\mathcal{Y}|$ is the number of labels.

Prompt-based methods Since our method is a brand-new basic prompt-learning method, our main purpose is to demonstrate its effectiveness compared to MLM-like methods, and we think it is not necessary to compare with more complex methods such as continuous prompt or automatic prompt methods. Therefore we choose token-level model PET (Schick and Schütze, 2021b,a) based on MLM and sentence-level model EFL[7] (Wang et al., 2021) based on entailment as two baselines.

4.3 Experiment Settings

Evaluation Protocol For few-shot learning, we follow the evaluation protocol adopted in (Gao et al., 2021; Liang et al., 2022) and assume K samples per class for training set. For English tasks the K of training set is set to 16, and the size of the development set is 10 times the size of the training set. The number K of FewCLUE has been set to 8 or 16 according to Xu et al. (2021). For each experiment, we run 5 experiments with 5 different training and development set (split by 5 fixed random seed) and report the average results and standard deviations.

Language Models In order to conduct comparative experiments fairly, for our main experiments, we use the same pre-trained language model for the same dataset. For English tasks, we adopt the vanilla English BERT-LARGE[8]. For Chinese tasks, we adopt the Chinese BERT-BASE[9] trained by UER using MLM and NSP (Zhao et al., 2019).

Hyper-parameters For few-shot learning, we train 10 epochs on all the datasets. We set learning rate as 2e-5 for English tasks, and 1e-5 for Chinese tasks. The batch size is 8. All baselines use the same hyper-parameters described above.

4.4 Main Results

The Table 1 reports the main results on 7 English and 4 Chinese single-sentence classification tasks.

[6] https://aistudio.baidu.com/aistudio/competition/detail/83

[7] We use MNLI(Williams et al., 2018) and OCNLI(Hu et al., 2020) to pre-train EFL.

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

[9] https://github.com/dbiir/UER-py

		Model	Corpus	English Tasks						
				SST-2	MR	CR	MPQA	Subj	Yahoo!	AGNews
Zero	PET	RoBERTa	$\mathcal{C}_{\mathrm{B}}$	81.2	75.6	76.6	63.3	**63.6**	18.7	47.8
			$\mathcal{C}_{\mathrm{R}}$	**83.6**	**80.8**	**79.5**	**67.6**	53.6	**25.6**	**54.5**
		BERT	$\mathcal{C}_{\mathrm{B}}$	67.6	65.3	61.2	63.9	**61.0**	25.6	**54.5**
			$\mathcal{C}_{\mathrm{B+Mix5}}$	**75.0**	**70.1**	**67.4**	**64.2**	55.3	**28.5**	38.4
	NSP-BERT	BERT	$\mathcal{C}_{\mathrm{B}}$	75.6	74.4	59.4	59.9	**53.9**	47.0	**77.5**
			$\mathcal{C}_{\mathrm{B+Mix5}}$	**81.2**	**78.3**	**76.9**	**72.4**	53.0	**56.8**	75.8
Few	PET	RoBERTa	$\mathcal{C}_{\mathrm{B}}$	88.6±1.5	83.9±0.8	87.8±0.7	82.0±1.1	82.8±5.6	65.2±1.3	86.0±0.4
			$\mathcal{C}_{\mathrm{R}}$	**91.7±0.6**	**88.0±0.5**	**91.5±0.9**	**85.6±2.1**	**87.8**±2.2	**68.9±1.0**	**87.8±0.9**
		BERT	$\mathcal{C}_{\mathrm{B}}$	85.3±1.7	80.3±2.1	89.2±0.3	83.3±2.4	85.4±1.9	64.3±1.3	84.0±1.0
			$\mathcal{C}_{\mathrm{B+Mix5}}$	**87.6±0.9**	**85.0±0.8**	**89.6±0.8**	**85.0±1.7**	**90.5±1.2**	**68.4**±0.7	**87.8±0.6**
	NSP-BERT	BERT	$\mathcal{C}_{\mathrm{B}}$	86.7±2.1	80.3±1.8	86.7±1.7	83.9±1.1	86.6±0.9	64.5±0.5	85.9±0.8
			$\mathcal{C}_{\mathrm{B+Mix5}}$	**89.4±0.7**	**83.3**±1.1	**88.7±1.0**	**85.3±1.0**	**92.1**±1.1	**68.3**±1.3	**87.6±0.5**

Table 2: Impact of pre-training corpus. $\mathcal{C}_{\mathrm{B}}$: pre-training from scratch with BERT's corpus; $\mathcal{C}_{\mathrm{R}}$: pre-training from scratch with RoBERTa's corpus; $\mathcal{C}_{\mathrm{B+Mix5}}$: continue pre-training with RoBERTa's corpus based on vanilla BERT.

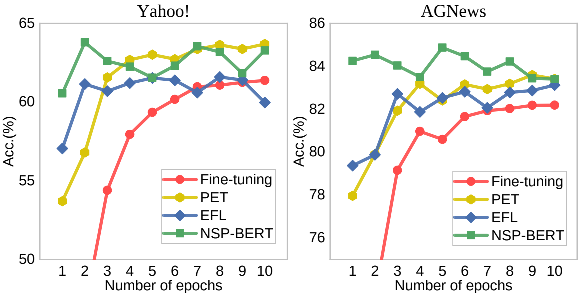

Figure 5: The accuracy of the 4 methods for each epoch during few-shot training on Yahoo! and AGNews.

	SST-2	MR	CR	MPQA
NSP-BERT	86.8±1.3	**80.5±1.5**	**86.0±2.2**	**83.9±1.1**
coupled→decouple	**86.8±1.2**	78.9±2.3	85.8±9.5	81.5±5.8
BCE→softmax	83.8±5.0	76.4±6.4	80.5±10.0	73.3±9.5
w/o NSP head	83.8±6.5	74.3±9.2	79.0±8.1	73.2±10.1
linear head+softmax	80.2±7.6	71.9±12.3	82.6±6.7	73.8±11.1

Table 3: Ablation studies of NSP-BERT on vanilla English BERT-Large. coupled→decouple: change coupled positive and negative samples to decoupled; BCE→softmax: change binary cross-entropy loss to softmax loss; w/o NSP head: use an initialized sigmoid head; linear head+softmax: use an initialized sigmoid head and softmax loss.

Since we use the same pre-trained language model for all methods, this experiment is fair enough. It is clear that our NSP-BERT offers distinct advantages in zero-shot scenario, particularly for multi-topic classification tasks such as Yahoo!, AGNews, and all Chinese datasets. In few-shot scenario. its performance is comparable to the MLM-based PET (Schick and Schütze, 2021a) on the most datasets. Compared with EFL (Wang et al., 2021) without pre-training on the NLI dataset, NSP-BERT is much better. Our NSP-BERT has the fastest convergence speed based on convergence curves, as shown in Figure 5. NSP-BERT usually achieves the best performance during the first few epochs.

Ablation studies on NSP-tuning It can be seen from Table 3 that coupling positive and negative samples + BCE loss function is the most effective and robust way of NSP-tuning. Other modifications in the table will degrade the performance of the model and make the results unstable. We believe this is due to the special output of the NSP Head, and re-initialization will lose the knowledge gained during pre-training.

Impact of Pre-training Corpus Compared with the RoBERTa model, the original BERT model has a large gap in the pre-training corpus. BERT is only pre-trained on Wikipedia and BookCorpus(Zhu et al., 2015), and the size is about 16GB, while RoBERTa additionally uses CC-News[10], OpenWebText (Gokaslan and Cohen, 2019) and Stories(Trinh and Le, 2018) corpus, which is 145GB more. We use the above 5 corpora[11] to pre-train the vanilla BERT model incrementally. Due to the limited computing power, our total training steps are about 30% of the BERT model and 2% of the RoBERTa model. As shown in Table 2, although it has not yet reached the level of RoBERTa, our BERT model ($\text{BERT}_{\mathcal{C}_{\mathrm{B+Mix5}}}$) has greatly improved the performance of zero-shot and few-shot learning, and this improvement even exceeds the changes brought by the prompt method.

[10] https://commoncrawl.org/2016/10/news-dataset-available/

[11] Since there is no public Stories corpus, we refer to the construction method of (Trinh and Le, 2018) and build it on the basis of CC-100 (Conneau et al., 2020).

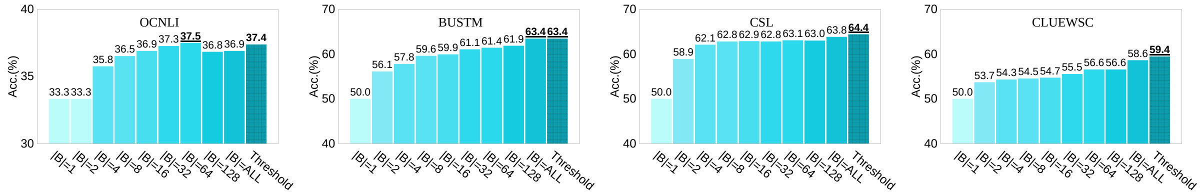

Figure 6: The performance of the samples-contrast answer mapping method under different preconditions on OCNLI, BUSTM, CSL and CLUEWSC. Batch size $|\mathcal{B}| \in \{1, 2, ..., 128, \mathrm{ALL}\}$, when the batch size is 1 (1 and 2 for OCNLI), the result is a random guess, when the batch size is ALL, indicating that the entire test set is obtained at one time. `Thresholds` means that the thresholds are obtained through the dev set, and then used for the prediction of the test set.

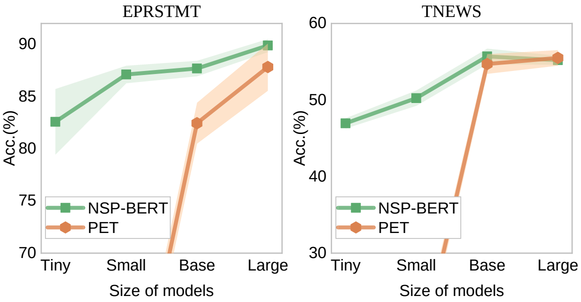

Figure 7: Accuracy of PET and NSP-BERT on EPRSTMT and TNEWS under 4 different model sizes.

Impact of Model Size Only under the premise of fixing the same pre-training corpus, we can verify the effect of model size on NSP-BERT. We carefully selected 4 sizes of UER's BERT (tiny, small, base and large) trained on same corpus for validation on two datasets, EPRSTMT and TNEWS. Figure 7 shows the impact of different sizes of models on NSP-BERT and PET, it can be seen that our method is still very competitive on small models[12].

4.5 Applications of NSP-BERT

We validate applications of NSP-BERT on the tasks shown in Table 4, including NLI (OCNLI, MNLI, SNLI, QNLI and RTE), text matching (BUSTM), keyword recognition (CSL), Chinese idiom cloze test (ChID), and coreference resolution (CLUEWSC). In these tasks, the zero-shot learning prediction ability of NSP-BERT is demonstrated with the help of the sample-contrast method. From Figure 6, we can see that even a small contrast batch size can help the sentence-pair tasks, and as the batch size increases, this improvement becomes more obvious and tends to be stable.

Our NSP-BERT can be applied to the task of entity typing, and can even handle entity linking task. The difficulty of entity linking for MLM-based model such as PET is that the description of the entity is of variable length. In these tasks with more than one target words or entity, the effect of two-stage prompt is obvious, see Table 5.

	Chinese Tasks				
	OCNLI	**BUSTM**	**CSL**	**WSC**	**ChID**
Majority	*38.1*	*50.0*	*50.0*	*50.0*	*14.3*
PET	**40.3**	50.6	52.2	54.7	**57.6**
NSP-BERT	37.4	**63.4**	**64.4**	**59.4**	52.0
	English Tasks				
	MNLI-m	**MNLI-mm**	**SNLI**	**QNLI**	**RTE**
Majority	*32.7*	*33.0*	*33.8*	*49.5*	*52.7*
PET	**47.1**	**46.0**	36.0	49.0	51.6
NSP-BERT	39.4	39.2	**43.4**	**67.6**	**55.6**

Table 4: Applications of NSP-BERT on FewCLUE tasks in zero-shot scenario. We report accuracy for all datasets. We only use the candidate-contrast method on ChID, and use the sample-contrast method on the rest of the datasets.

	DuEL2.0-L	**DuEL2.0-T**
PET	-	40.0
NSP-BERT	61.2 / 69.7↑	31.4 / 40.0↑

Table 5: Word sense disambiguation task. DuEL2.0-L: DuEL2.0 entity linking part; DuEL2.0-T: DuEL2.0 entity typing part. The left side of the slash is the one-stage prompt, and the right side is the two-stage prompt.

5 Conclusion

In this paper, we show that NSP can also be an apposite zero-shot or few-shot learner same as MLM. This not only provides a new route for prompt-learning, but also makes us rethink the role of sentence-level pre-training tasks. At the same time, we continue to pre-train the BERT model with a small amount of computing power, and its performance improves significantly on both zero-shot

[12] PET fails to fit on tiny and small models for no reason.

and few-shot learning, whether to use PET or NSP-BERT. We believe that not only the size of the model, but also the pre-training corpus, both determine the upper limit of the model's ability on few-shot learning.

6 Acknowledgements

We thank Su Jianlin for the repository bert4keras and the spirit of open source. We thank the insightful comments from the anonymous reviewers.

References

Leonard Adolphs, Shehzaad Dhuliawala, and Thomas Hofmann. 2021. How to query language models?

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. of EMNLP*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proc. of NeurIPS*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proc. of NeurIPS*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. In *Proc. of ACL*.

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

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proc. of ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proc. of NeurIPS*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proc. of ACL*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Hai Hu, Kyle Richardson, Xu Liang, Li Lu, Sandra Kübler, and Larry Moss. 2020. OCNLI: Original Chinese natural language inference. In *Findings of Empirical Methods for Natural Language Processing (Findings of EMNLP)*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proc. of KDD*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *Proc. of EMNLP*.

LTD. IFLYTEK CO. 2019. Iflytek: a multiple categories chinese text classifier. *competition official website, http://challenge.xfyun.cn/2019/gamelist*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. In *Proc. of EMNLP*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Xiaozhuan Liang, Ningyu Zhang, Siyuan Cheng, Zhen Bi, Zhenru Zhang, Chuanqi Tan, Songfang Huang, Fei Huang, and Huajun Chen. 2022. Contrastive demonstration tuning for pre-trained language models.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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*.

Conversational-AI Center of OPPO XiaoBu. 2021. Bustm: Oppo xiaobu dialogue short text matching dataset.

Bo PANG. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proc. of ACL*.

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL*.

Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases. In *Proc. of EMNLP*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. of EACL*.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proc. of NAACL*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proc. of EMNLP*.

Livio Baldini Soares, Nicholas Arthur FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proc. of ACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.

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 preprint arXiv:1909.00204*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proc. of CIKM*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. Clue: A chinese language understanding evaluation benchmark. In *Proc. of COLING*.

Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Hu Yuan, Huilin Xu, Guoao Wei, Xiang Pan, and Hai Hu. 2021. Fewclue: A chinese few-shot learning evaluation benchmark.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proc. of NeurIPS*.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. In *Proc. of ICLR*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. In *Proc. of EMNLP*.

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

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proc. of NAACL*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Models

A.1 Probability Formula

We compared the output probability formulas of different zero-shot prompt-learning models include our NSP-BERT. The following description is a general situation, assuming that each label it mapped to a span with a length is greater than or equal to 1. When the length of the label word is equal to 1, the form of the pre-training and downstream tasks tend to be unified. When the length is greater than 1, there is a gap between them, even we use the model pre-trained by whole word masking (Cui et al., 2019) or span masking (Joshi et al., 2020).

PET-Zero Denote the token in position i as t_i, the label span will be replaced by $\texttt{[MASK]}_{l:r}$. When ignoring special tokens such as `[CLS]` and `[PAD]`, the input of PET-Zero is:

$$\mathbf{x}_{input} = t_1, ..., \texttt{[MASK]}_l, ..., \texttt{[MASK]}_r, ... \tag{4}$$

The output probability for label $y_i^{(j)}$ is:

$$q(y_i^{(j)}|\mathbf{x}_i) = \underset{1 \leqslant j \leqslant M}{\mathrm{softmax}}\Big(\prod_{l \leqslant v \leqslant r} q_{\mathcal{M}_{\mathrm{MLM}}}(t_v^{(j)}|\mathbf{x}_{input})\Big). \tag{5}$$

NSP-BERT For our NSP-BERT, the label span $t_{l:r}^{(j)}$ will be replaced in turn:

$$\mathbf{x}_{input}^{(j)} = t_1, ..., \texttt{[SEP]}, ..., t_l^{(j)}, ..., t_r^{(j)}, ... \tag{6}$$

The output probability for label $y_i^{(j)}$ is:

$$q(y_i^{(j)}|\mathbf{x}_i) = \underset{1 \leqslant j \leqslant M}{\mathrm{softmax}}(q_{\mathcal{M}_{\mathrm{NSP}}}(\mathbf{x}_{input}^{(j)})). \tag{7}$$

A.2 Parameters of Models

For FewCLUE, we use the Chinese vanilla-BERT-Base pre-trained by UER (Zhao et al., 2019) for the main results of our NSP-BERT. We also report the results of the other scales (tiny, small and large) model. Following the implementation of (Xu et al., 2021), we use Chinese RoBERTa-wwm-ext-Base pre-trained by HFL (Cui et al., 2019) and NEZHA-Gen (Wei et al., 2019) for the baselines.

For English datasets, following the implementation [13] of (Gao et al., 2021). We use vanilla-BERT-Large pre-trained by Google (Devlin et al., 2018) for our NSP-BERT, and RoBERTa-Large[14] for the baselines.

Table 6 shows the hyperparameters of the models used in our experiment. The English and Chinese models are a little different in total parameters, mainly due to the different vocabulary size. It should be noted that not all pre-trained models fully stored NSP head and MLM head, so we need to select deliberately.

Model	L	H	A	**Total Parameters** ZH	 / EN
RoBERTa	12	768	12	102M	-
RoBERTa-Large	12	768	12	-	355M
BERT-Tiny	3	384	6	14M	-
BERT-Small	6	512	8	31M	-
BERT-Base	12	768	12	102M	-
BERT-Large	24	1024	16	327M	355M

Table 6: The parameters of different models used in our experiment. L: number of layers; H: hidden size; A: number of self-attention heads; "-": not used in our paper; ZH: Chinese model; EN: English model.

A.3 Others

Marks and Two-stage prompt In the Figure 8, we compare the markers that usually appear in supervised training (Huang et al., 2019; Soares et al., 2019; Wu and He, 2019; Zhong and Chen, 2021). The marker are special tokens such as `[noun]`, `[pron]` and `[e]`. They are usually added before and after the target words. The two-stage prompt plays the same role as the markers, but it uses a natural language description method.

B More Details

B.1 Datasets

FewCLUE FewCLUE (Xu et al., 2021) is a Chinese few-shot learning evaluation benchmark with 9 Chinese NLU tasks in total. There are 4 single-sentence tasks which are EPRSTMT, TNEWS, CLSDCP and IFLYTEK. EPRSTMT is a binary sentiment analysis dataset for E-commerce reviews. TNEWS (Xu et al., 2020) is a short text classification for news title with 15 topics. CSLDCP is a text classification dataset including abstracts from a variety of Chinese scientific papers and with 67 categories in total. IFLYTEK (IFLYTEK CO., 2019) is a long text classification dataset for App descriptions. There are 3 sentence-pair tasks which are OCNLI, BUSTM and CSL. OCNLI (Hu et al., 2020) is an original Chinese NLI tasks. BUSTM (of OPPO XiaoBu, 2021) is a dialogue short text

[13] https://github.com/princeton-nlp/LM-BFF

[14] https://github.com/pytorch/fairseq/tree/main/examples/roberta

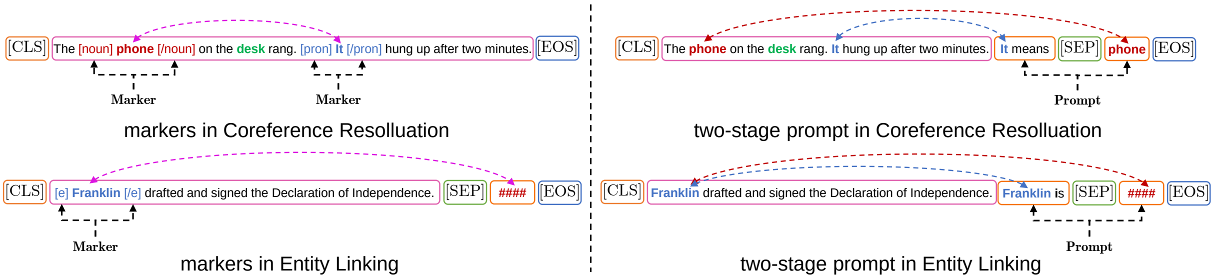
[CLS] The [noun] phone [/noun] on the desk rang. [pron] It [/pron] hung up after two minutes. [EOS]
Marker
Marker
markers in Coreference Resolluation
[CLS] The phone on the desk rang. It hung up after two minutes. It means [SEP] phone [EOS]
Prompt
two-stage prompt in Coreference Resolluation
[CLS] [e] Franklin [/e] drafted and signed the Declaration of Independence. [SEP] #### [EOS]
Marker
markers in Entity Linking
[CLS] Franklin drafted and signed the Declaration of Independence. Franklin is [SEP] #### [EOS]
Prompt
two-stage prompt in Entity Linking

Figure 8: The comparison of markers (Left) and two-stage prompt (Right), examples in coreference resolution and entity linking/typing tasks.

matching task. CSL is a abstract-keywords matching task. There are other two tasks ChID and CLUEWSC. ChID (Zheng et al., 2019) is a Chinese idiom cloze test dataset. CLUEWSC is a coreference resolution task.

For all the datasets in FewCLUE, we evaluate our model on the public test set. Although FewCLUE provides a large number of unlabeled samples, we did not use them in the our experiment, so the results are unable to be compared with the results on the leaderboard[15]. For dataset TNEWS, we did not use the information of keywords following (Xu et al., 2021). We treat CLUEWSC as a sentence-pair task due to its data characteristics.

DuEL2.0 We divide DuEL2.0 into two parts. In the first part, the entity linking part, there are 26586 samples. All the samples' mention can be mapped to single or multiple entities in the knowledge base, and each mention can be linked to 5.37 entities on average. In the second part, the entity typing part, there are 6465 samples. Those samples' mention cannot be found in the knowledge base, but they will be divided into their corresponding upper entity types. There are a total of 24 upper entity types, and we do not remove the `Other` type. When performing the entity linking part, we only use the entity's summary information, without using more entity triples.

Entity Linking	Ave. Entities	Entity Tpying	Types
26586	5.37	6465	24

Table 7: Since the DuEL2.0's test set is not public, we use the dev set to test our model. The the number of the original text lines is 10000. According to the predicted target (entities in knowledge base or upper types), we manually divide it into two parts, entity linking and entity typing.

English Datasets Following (Gao et al., 2021; Hu et al., 2021; Liang et al., 2022), we evaluate our model on 7 single-sentence and 5 sentence-pair English tasks. For the datasets SST-2 (Socher et al., 2013), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005; Bar Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), we follow (Gao et al., 2021) and (Zhang et al., 2021) and use their original development sets for testing. For datasets MR (PANG, 2005), CR (Hu and Liu, 2004), MPQA (Wiebe et al., 2005), Subj (Pang and Lee, 2004), Yahoo! and AGNews(Zhang et al., 2015), we use the testing set randomly sampled from training set and leaved from training by (Gao et al., 2021)[16]. For SNLI (Bowman et al., 2015), we use their official test sets.

B.2 Results

Different Templates We compared in detail the performance of NSP-BERT under different prompt templates. This experiment wad conducted on 4 Chinese single-sentence classification datasets.

- **Template 1** uses just the original label words.
- **Template 2** adds pronouns and copulas such as "I am", "it is" or "this is", to make the template become a complete sentence.
- **Template 3** incorporates more domain information into the prompts, such as "shopping", "news", "paper" and "app". This makes the original input sentence and prompt have better connectivity.

For zero-shot learning, the prompt templates have a strong impact on the performance, and for different models, there is a big difference. Therefore, we verified the influence of templates for different models versions and scales. The results are shown in Table 9, Table 10, Table 11 and Table 12.

[15] https://www.cluebenchmarks.com/fewclue.html

[16] https://nlp.cs.princeton.edu/projects/lm-bff/datasets.tar

Category	Corpus	#Train	#Test	$\lvert\mathcal{Y}\rvert$	Task Type	Metrics	Source
				English Tasks			
Single-Sentence	SST-2	6,920	872	2	Sentiment Analysis	Acc.	Movie Reviews
	MR	8,662	2,000	2	Sentiment Analysis	Acc.	Movie Reviews
	CR	1,775	2,000	2	Sentiment Analysis	Acc.	E-commerce Reviews
	MPQA	8,606	2,000	2	Opinion Polarity	Acc.	World Press
	Subj	8,000	2,000	2	Subjectivity	Acc.	Movie Reviews
	Yahoo!	1,400,000	6,000	10	Question Classification	Acc.	Yahoo
	AGNews	8,551	7,600	4	News Topic Classification	Acc.	Web
Sentence-Pair	MNLI	392,702	9,815	3	Natural Language Inference	Acc.	Speech, Fiction and Reports
	MNLI-mm	392,702	9,832	3	Natural Language Inference	Acc.	Speech, Fiction and Reports
	SNLI	549,367	9,842	3	Natural Language Inference	Acc.	Image Captions
	QNLI	104,743	5,463	2	Natural Language Inference	Acc.	Wikipedia
	RTE	2,490	277	2	Natural Language Inference	Acc.	News and Wikipedia
				Chinese Tasks (FewCLUE)			
Single-Sentence	EPRSTMT	32	610	2	Sentiment Analysis	Acc.	E-commerce Reviews
	TNEWS	240	2,010	15	Short Text Classification	Acc.	News Title
	CSLDCP	536	1,784	67	Long Text Classification	Acc.	Academic CNKI
	IFLYTEK	928	1,749	119	Long Text Classification	Acc.	App Description
Sentence-Pair	OCNLI	32	2,520	3	Natural Language Inference	Acc.	5 genres
	BUSTM	32	1,772	2	Short Text Matching	Acc.	AI Virtual Assistant
	CSL	32	2,828	2	Keyword Recognition	Acc.	Academic CNKI
Others	ChID	42	2,002	7	Chinese Idiom Cloze Test	Acc.	Novel, Essay News
	CLUEWSC	32	976	2	Coreference Resolution	Acc.	Chinese Fiction Books

Table 8: Task descriptions and statistics. In FewCLUE we omit the unlabeled dataset because it is not used. Test of FewCLUE indicates the number of samples in the public test set. The 5 text genres of OCNLI are government documents, news, literature, TV talk shows and telephone conversations.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
UER	BERT-Tiny	68.13/76.56	75.00/80.82	**81.88/80.33**
	BERT-Small	85.00/87.70	82.50/87.70	**87.50/86.72**
	BERT-Base	60.00/54.59	78.75/80.98	**88.13/86.89**
	BERT-Large	78.13/82.79	83.75/82.62	**84.38/84.43**

Table 9: Zero-shot acc. of NSP-BERT on EPRSTMT.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
UER	BERT-Tiny	38.80/36.62	39.25/36.37	**41.07/38.56**
	BERT-Small	38.98/38.81	39.80/40.35	**41.80/42.19**
	BERT-Base	41.26/41.84	46.99/48.66	**50.64/51.00**
	BERT-Large	45.17/42.79	48.72/48.31	**54.28/53.83**

Table 10: Zero-shot acc. of NSP-BERT on TNEWS.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
UER	BERT-Tiny	24.03/25.73	**27.37/29.60**	25.68/28.81
	BERT-Small	28.48/30.72	29.35/31.45	**29.78/31.78**
	BERT-Base	39.80/40.53	44.87/45.80	45.26/**47.59**
	BERT-Large	44.73/42.83	44.00/44.34	**45.89**/46.92

Table 11: Zero-shot acc. of NSP-BERT on CSLDCP.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
UER	BERT-Tiny	32.70/32.65	31.97/34.13	**33.65/34.59**
	BERT-Small	32.27/32.42	**35.54**/34.65	35.25/**34.76**
	BERT-Base	36.41/36.59	42.39/40.19	**43.12/41.62**
	BERT-Large	37.73/36.94	44.28/**42.60**	**44.87**/42.42

Table 12: Zero-shot acc. of NSP-BERT on IFLYTEK.

Probability of NSP in sentence-pair tasks To further explain the necessity for us to propose sample-contrast mapping method, we show the NSP output probability of the sentence-pair tasks in Figure 9 and Figure 10. It's not difficult to see that the NSP probability of most samples is close to 1. So we can not judge its label for a individual sample. We need to contrast different samples, and predict the label by obtaining the distribution of the dataset.

Impact of batch size for samples-contrast In one case, we cannot get the entire test set at once, then we need to predict the samples of the test set batch by batch. We set the batch size $|B| \in \{1, 2, ..., 128, \text{ALL}\}$, to observe the results predicted by samples-contrast method (see Table 13). As the batch size increases, the performance improves and stabilizes. Of course, when the batch size is less than the number of labels, the result is equivalent to random guessing. In another case, we cannot get the distribution of the test set, that is, we don't know the proportion of each label. Then we can use the development to calculate the NSP probability threshold of each label to predict the test set. The model can also get the desired performance.

Strategies for datasets For different datasets, according to their characteristics, the position of the prompt (prefix or suffix), and the mapping method (candidates-contrast or samples-contrast) are different. We take Chinese tasks as examples, all the strategies are shown in Table 14. In the single-sentence classification tasks (EPRSTMT, TNEWS, CSLDCP, IFLYTEK), the prompts are all prefixed, and we adopt candidates-contrast. For the word sense disambiguation tasks (CLUEWSC and DuEL2.0), since we need to utilize two-stage prompt method, we all use the suffix. In sentence-pair tasks (OCNLI, BUSTM and CSL), we choose the appropriate order through the development set to arrange the two sentences, where suffix means using the original order and prefix means using the reverse order.

Prompts for datasets Due to the number of data sets in our paper, we report in detail the prompt templates of the more important Chinese datasets in Table 16, and briefly report the prompts of English datasets in Table 15.

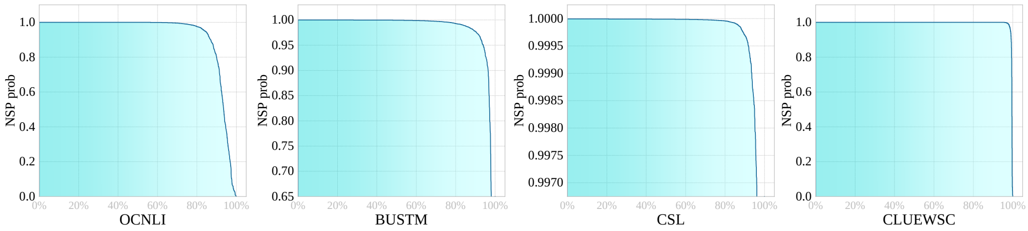

Figure 9: The NSP output probability of the 4 sentence-pair tasks OCNLI, BUSTM, CSL and CLUEWSC in Chinese benchmark FewCLUE. The x-axis represents the proportion of the samples. And the y-axis represents the NSP probability of the samples.

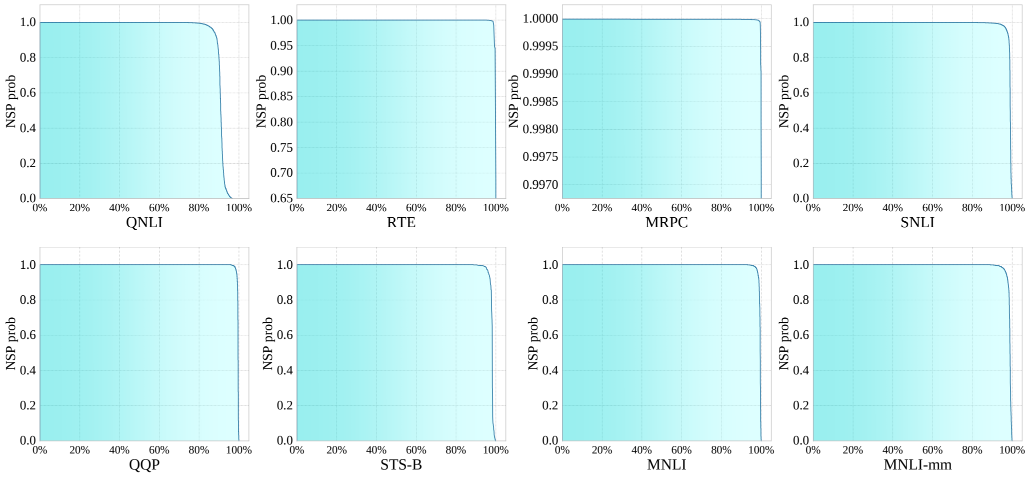

Figure 10: The NSP output probability of the 8 English sentence-pair tasks QNLI, RTE, MRPC, SNLI, QQP, STS-B, MNLI and MNLI-mm. The x-axis represents the proportion of the samples. And the y-axis represents the NSP probability of the samples.

Dataset	**Dev**	**Test**									
		$\vert\mathcal{B}\vert=1$	$\vert\mathcal{B}\vert=2$	$\vert\mathcal{B}\vert=4$	$\vert\mathcal{B}\vert=8$	$\vert\mathcal{B}\vert=16$	$\vert\mathcal{B}\vert=32$	$\vert\mathcal{B}\vert=64$	$\vert\mathcal{B}\vert=128$	$\vert\mathcal{B}\vert=$All	Threshold
OCNLI	37.50	33.33	33.33	35.75	36.51	36.90	37.26	**37.50**	36.83	36.90	37.38
BUSTM	62.50	50.00	56.09	67.79	59.59	59.93	61.06	61.40	61.85	**63.43**	**63.43**
CSL	64.38	50.00	58.91	62.09	62.79	62.86	62.79	63.07	63.00	63.85	**64.41**
CLUEWSC	57.23	50.00	53.69	54.30	54.51	54.71	55.53	56.56	56.56	58.61	**59.43**
MNLI-m	41.67	35.22	35.22	39.08	**40.04**	39.08	39.63	39.33	39.48	39.33	39.41
MNLI-mm	39.58	35.45	35.45	38.41	38.59	38.62	38.19	37.69	38.24	38.17	**39.17**
SNLI	43.75	34.28	34.28	**44.14**	44.21	43.54	43.20	43.17	43.13	43.35	43.42
QNLI	87.50	49.46	62.37	64.63	65.37	66.58	66.87	67.23	67.34	67.56	**67.56**
RTE	62.50	52.71	52.71	54.87	53.43	55.60	54.15	**54.15**	54.87	51.99	55.60

Table 13: The performance of the samples-contrast answer mapping method under different preconditions on sentence-pair tasks. Batch size $|\mathcal{B}| \in \{1, 2, ..., 128, \mathrm{ALL}\}$, when the batch size is less than the number of labels, the result is a random guess, when the batch size is ALL, indicating that the entire test set is obtained at one time. `Thresholds` means that the thresholds are obtained through the development set, and then used for the prediction of the test set.

Strategies		Single-Sentence Task				Sentence-Pair Task			Others		DuEL2.0	
		EPRSTMT	TNEWS	CSLDCP	IFLYTEK	OCNLI	BUSTM	CSL	ChID	CLUEWSC	Entity Linking	Entity Typing
Prompt	Prefix	✓	✓	✓	✓	✓	✓					
	Suffix							✓	✓	✓	✓	✓
Answer	C-C	✓	✓	✓	✓				✓		✓	✓
Mapping	S-C					✓	✓	✓		✓		

Table 14: Strategies adopted on the 10 datasets in FewCLUE and DuEL2.0. The **prefix** means to put the prompt in front of the original text, and the **suffix** is the opposite. **C-C** means candidates-contrast answer mapping method, and **S-C** means samples-contrast answer mapping method.

Task	Method	Prompt Templates
SST-2		Original Labels: negative; positive
	PET	Mapping Words: terrible; great Prompt Template: x It was `[label]`.
	NSP-BERT	Mapping Words: terrible; great Prompt Template: A `[label]` piece of work `[SEP]` x
MR		Original Labels: negative; positive
	PET	Mapping Words: terrible; great Prompt Template: x It was `[label]`.
	NSP-BERT	Mapping Words: terrible; great Prompt Template: A `[label]` piece of work `[SEP]` x
CR		Original Labels: negative; positive
	PET	Mapping Words: terrible; great Prompt Template: x It was `[label]`.
	NSP-BERT	Mapping Words: terrible; great Prompt Template: It was `[label]. [SEP]` x
Subj		Original Labels: negative; positive
	PET	Mapping Words: subjective; objective Prompt Template: x This is `[label]`.
	NSP-BERT	Mapping Words: subjective; objective Prompt Template: A `[label]` comment `[SEP]` x
MPQA		Original Labels: negative; positive
	PET	Mapping Words: terrible; great Prompt Template: x It was `[label]`.
	NSP-BERT	Mapping Words: negative; positive Prompt Template: It is `[label]. [SEP]` x
Yahoo!		Original Labels: Society & Culture; Science & Mathematics; Health; Education & Reference; Computers & Internet; Sports; Business & Finance; Entertainment & Music; Family & Relationships; Politics & Government
	PET	Mapping Words: Society; Science; Health; Education; Computer; Sports; Business; Entertainment; Relationship; Politics Prompt Template: `[label]` question: x
	NSP-BERT	Mapping Words: Society; Science; Health; Education; Computer; Sports; Business; Entertainment; Relationship; Politics Prompt Template: `[label]` question: `[SEP]` x
AGNews		Original Labels: political; sports; business; technology
	PET	Mapping Words: political; sports; business; technology Prompt Template: A `[label]` news : x
	NSP-BERT	Mapping Words: political; sports; business; technology Prompt Template: A `[label]` news : `[SEP]` x

Table 15: The prompts used in English datasets. We only show the template with best performance. We select the most suitable prompt template for PET and NSP respectively. `[label]` is the token will be replaced by the mapping words. EFL(Wang et al., 2021) uses the exact same prompts as NSP-BERT.

Task	Prompt Templates	Mapping words of PET	Mapping words of NSP-BERT
EPRSTMT	Template 1: $\mathbf{x}$ [SEP] 很[label]. Template 2: $\mathbf{x}$ [SEP] 东西很[label]. Template 3: $\mathbf{x}$ [SEP] 这次买的东西很[label]	好; 差	好; 差
TNEWS	Template 1: $\mathbf{x}$ [SEP] [label]. Template 2: $\mathbf{x}$ [SEP] [label]新闻. Template 3: $\mathbf{x}$ [SEP] 这是一则 [label]新闻.	故事; 文化; 娱乐; 体育; 财经; 房产; 汽车; 教育; 科技; 军事; 旅游; 国际; 股票; 农业; 电竞	故事; 文化; 娱乐; 体育; 财经; 房产; 汽车; 教育; 科技; 军事; 旅游; 国际; 股票; 农业; 电竞
CSLDCP	Template 1: $\mathbf{x}$ [SEP] [label]. Template 2: $\mathbf{x}$ [SEP] [label] 论文. Template 3: $\mathbf{x}$ [SEP] 这是一篇 [label]论文.	材料; 作物; 口腔; 药学; 教育; 水利; 理经; 食品; 兽医; 体育; 核能; 力学; 园艺; 水产; 法学; 地质; 能源; 农林; 通信; 情报...	材料科学与工程; 作物学; 口腔医学; 药学; 教育学; 水利工程; 理论经济学; 食品科学与工程; 畜牧学/兽医学; 体育学; 核科学与技术; 力学; 园艺学...
IFLYTEK	Template 1: $\mathbf{x}$ [SEP] [label]. Template 2: $\mathbf{x}$ [SEP] [label] 类软件. Template 3: $\mathbf{x}$ [SEP] 这是一款 [label] 类软件.	打车; 地图; 免费; 租车; 同城; 快递; 婚庆; 家政; 交通; 政务; 社区; 赚钱; 魔幻; 仙侠; 卡牌; 飞行; 射击; 休闲; 动作; 体育; 棋牌; 养成; 策略; 竞技; 辅助...	打车; 地图导航; 免费WIFI; 租车; 同城服务; 快递物流; 婚庆; 家政; 公共交通; 政务; 社区服务; 薅羊毛; 魔幻; 仙侠; 卡牌; 飞行空战; 射击游戏; 休闲益智; 动作类; 体育竞技...

Table 16: The prompts used for single-sentence classification tasks in FewCLUE. [label] is the token will be replaced by the mapping words. The mapping words of PET need to be manually converted to equal length. Since there are two options for the prompt, **prefix** and **suffix**, we select the most suitable one through the development set. For dataset with a lot of labels, due to space considerations, we have omitted some of them.