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

Anonymous ACL submission

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 all prompt-based methods are token-level, meaning they all utilize GPT's left-to-right language model or BERT's masked language model to perform cloze-style tasks. In this paper, we attempt to accomplish several NLP tasks in the zero-shot scenario 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. Based on the characteristics of NSP-BERT, we offer several quick building templates for various downstream tasks. We suggest a two-stage prompt method for word sense disambiguation tasks in particular. Our samples-contrast method for mapping the labels significantly enhance the model's performance on sentence-pair tasks. On the Chinese benchmark FewCLUE, our NSP-BERT outperforms other zero-shot methods on most of these tasks and comes close to the few-shot methods. And on GLUE and other English datasets NSP-BERT is still competitive. Our code will be available on github.

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).

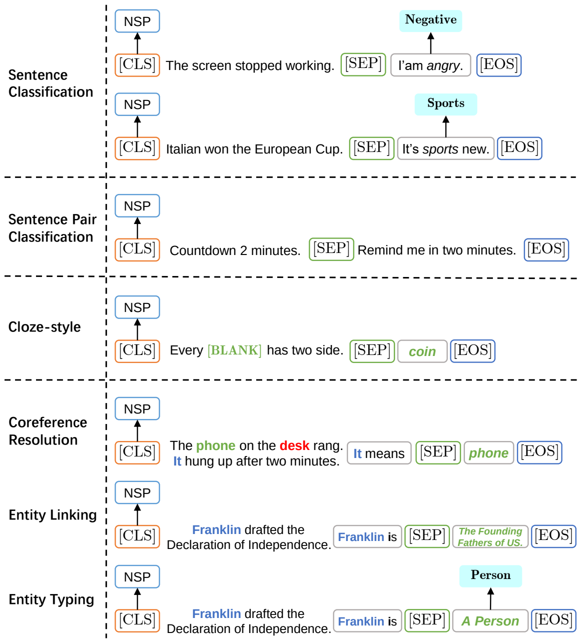

Figure 1: Prompts for various NLP tasks of NSP-BERT.

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.

The input text for the classification task, for example, "*The Italian team won the European Cup.*",

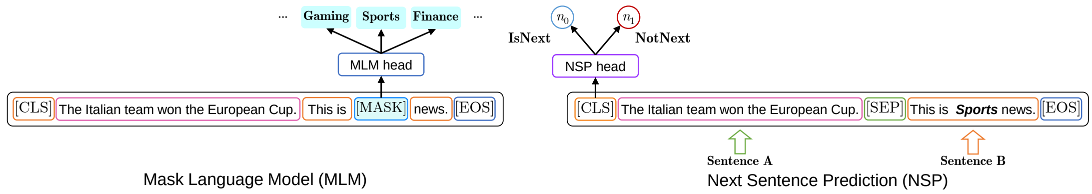

Figure 2: (Left) MLM task for token-level prompt-learning. (Right) NSP task for sentence-level prompt-learning.

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. On the Chinese benchmark FewCLUE, NSP-BERT has achieved the SOTA performance among zero-shot models without using any task-specific training data. Its performance is comparable to that of several few-shot learning methods. In English tasks such as GLUE, NSP-BERT still has strong competitiveness.
- Although the NSP probabilities of most sentence pairs are close to 1, we propose the samples-contrast method, which enables NSP-BERT to solve the sentence-pair task unsupervised.
- We suggest to use two-stage prompt construction methods to alleviate the problem that sentence-level prompt-based models are not sensitive to token positions, which further improves the performance of NSP-BERT on word sense disambiguation tasks.

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 sentiment classification input/label pair is reformulated to "**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 "**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 not on Zero-shot tasks unless it is trained 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, with P-tuning, GPTs achieve comparable and sometimes better performance to similar-sized BERTs in supervised learning.

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) pointed out that there are 3 types of bias (majority label bias, recency bias and common token bias) in GPT. By using content-free inputs (e.g. "N/A") to calibrate the model's output probabilities, the performance of GPT-2 and GPT-3 has been substantially improved.

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. $\mathbf{s} = \mathbf{W}_{\text{nsp}}\mathbf{h}_{\text{[CLS]}}$, where $\mathbf{h}_{\text{[CLS]}}$ is the hidden vector of 1 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

$n \in \{\texttt{IsNext}, \texttt{NotNext}\}$.

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

NSP is a self-supervised task that is simple and weak. We believe the task is more likely to judge whether two sentences are from the same document since the negative sample is randomly picked from another unrelated document. In other words, rather than determining the order of two phrases, the NSP task may determine if they have the same topic and express the same semantics.

The NSP task is quite similar to a contrastive learning task, as shown in Figure 3. So, does the NSP just compare sentence similarities or does it have the ability to reason logically? The following are the major reasons why we believe NSP has logical reasoning ability:

- **The NSP task is interactive.** Tokens in one sentence could interact with their own tokens while also interacting with tokens in the other sentence.
- **The NSP task is trained alongside the MLM task.** The MLM task provides a training basis for the self-attention mechanism of the entire model.

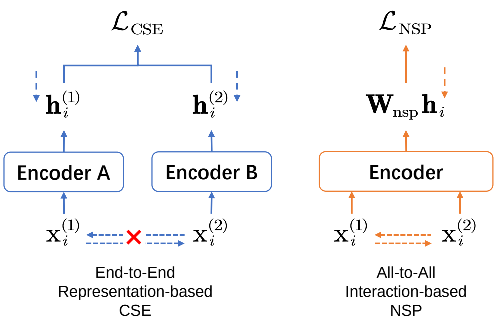

Figure 3: Conceptual comparison between End-to-End representation-based contrastive learning of sentence embeddings (CSE) and All-to-All interaction-based next sentence prediction (NSP). Except that the output of the model is not the representation of the sentence, the NSP task uses a weak self-supervision method to train the BERT.

NSP-BERT is a true prompt-based learner, not a sentence similarity matcher, as determined by the above two points. This will be confirmed in our experiments. The model performs better the closer the template is to a fluent and logical natural language sentence.

3.2 Prompts in NSP-BERT

NSP-BERT, like other prompt-based learning methods, requires the construction of appropriate templates for various tasks. Since NSP-BERT does not rely on the training data of any downstream tasks, 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.

Single-Sentence Task Samples must be classified into different topics in the single-sentence task. Suppose that the training dataset of a single-sentence classification task $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, $\mathbf{x}_i$ is the ith sentence in the total N samples, and the label of $\mathbf{x}_i$ is y_i, which can be mapped to $y^{(j)} \in \mathcal{Y}$, where $|\mathcal{Y}|$ is the number of topics in this dataset. For each $y^{(j)}$, it will be mapped to a prompt template $p^{(j)} \in \mathcal{P}$, $\mathcal{P}$ is the template sets. And the input of the model will be,

$$\mathbf{x}_{input} = \texttt{[CLS]}\,\mathbf{x}_i\,\texttt{[SEP]}\,p^{(j)}\,\texttt{[CLS]},$$

the probability when the label of sample $\mathbf{x}_i$ is $y^{(j)}$ is:

$$q(y^{(j)}|\mathbf{x}_i) = \frac{\exp q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p^{(j)})}{\sum_{p^{(k)} \in \mathcal{P}} \exp q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p^{(k)})}. \quad (2)$$

Sentence-Pair Task The sentence-pair tasks aim to identify the relationship between two sentences. This type of dataset $\mathcal{D} = \{(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, y_i)\}_{i=1}^{N}$ contains N samples, each with 2 sentences $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$. The relationship between them is y_i, which can be mapped to $y^{(j)} \in \mathcal{Y}$, where $|\mathcal{Y}|$ is the number of relationship types. The output of the NSP model $q_{\mathcal{M}}(\mathbf{x}_i)$ is shown in Eq. 3. (We do not directly associate the output of the NPS model directly with the labels here.)

$$q(\mathbf{x}_i) = q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}) \quad (3)$$

Cloze-Style Task The cloze-style task is to give a sentence with blanks, and the model must find the most appropriate tokens or spans to fill in the blanks. The dataset $\mathcal{D} = \{(\mathbf{x}_i, c_i^{(1)}, ..., c_i^{(j)}, ..., y_i)\}_{i=1}^{N}$. For each sample, there is a sentence $\mathbf{x}_i$ with a $\texttt{[BLANK]}$, and there are $|\mathcal{Y}_i|$ candidates $\{c_i^{(j)}\}_{j=1}^{|\mathcal{Y}_i|}$ to be chosen. For each option $c_i^{(j)}$, there is a template $p_i^{(j)} \in \mathcal{P}_i$ corresponding to it. Given the input:

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

the output of model is:

$$q(y_i^{(j)}|\mathbf{x}_i) = \frac{\exp q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p_i^{(j)})}{\sum_{p_i^{(k)} \in \mathcal{P}_i} \exp q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p_i^{(k)})}. \quad (4)$$

Word Sense Disambiguation In a fully supervised training scenario, we may add markers before and after the word to identify the word to be disambiguated (Huang et al., 2019; Soares et al., 2019; Wu and He, 2019) (See Appendix 8 for detailed comparison). Because there is no downstream tasks training data for our model, it is impossible to identify the target word's position by markers. We propose a **Two-Stage Prompt** construction method to indicate the target word using natural language descriptions in our NSP-BERT, as shown in Figure 4.

- **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.

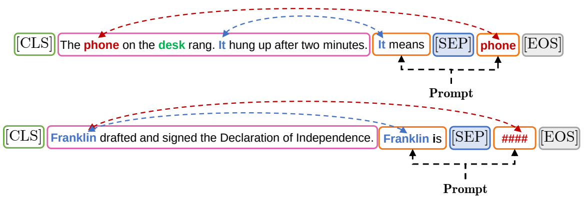

Figure 4: Two-stage prompt, examples in coreference resolution and entity linking/typing tasks.

Feed the two-stage prompt into the language model, and it will determine if the sentence is fluent and reasonable. 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:

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

3.3 Answer Mapping

It's easy to observe that not all probability outputs in the above tasks are directly linked with labels. This is because not all datasets can provide contrastive candidate objections (sentiments/topics/idioms/entities). Pre-trained language models, on the other hand, are not susceptible to negative inference (Kassner and Schütze, 2020), the NSP model is no exception. As a result, we propose two answer mapping methods, **candidates-contrast** answer mapping and **samples-contrast** answer mapping, for different situations.

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). As show in Figure 5. We take the highest probability output by $\mathcal{M}$ among the candidates as the final output answer where the condition is $\texttt{IsNext}$:

$$\begin{aligned} \hat{y}_i &= \arg\max_j q(y_i^{(j)}|\mathbf{x}_i) \\ &= \arg\max_j q_{\mathcal{M}}(n = \texttt{IsNext}|\mathbf{x}_i, p_i^{(j)}) \end{aligned} \tag{5}$$

Samples-Contrast For sentence-pair tasks, the NSP 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 5), to determine the label of a individual sample by contrast the probability of NSP between samples. To put it simply, by **rank**ing[1] in ascending order, the samples with a relatively higher NSP probability are **divide**d[2] into labels with a higher degree of matching, such as $\texttt{Entailment}$. On the contrary, samples with lower NSP probability will be divided to labels such as $\texttt{NotEntailment}$. This procedure is summarized in Algorithm 1.

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.

4 Experiment

4.1 Tasks and Datasets

FewCLUE We evaluate our model mainly on 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. The number of training samples per class K is setted to 8 or 16. See details in Appendix B.1.

DuEL2.0 In order to further verify the ability of NSP-BERT for word sense disambiguation, the entity linking dataset DuEL2.0[3] was added. And

[1] Sort samples in ascending or descending order according to NSP probability.

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

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

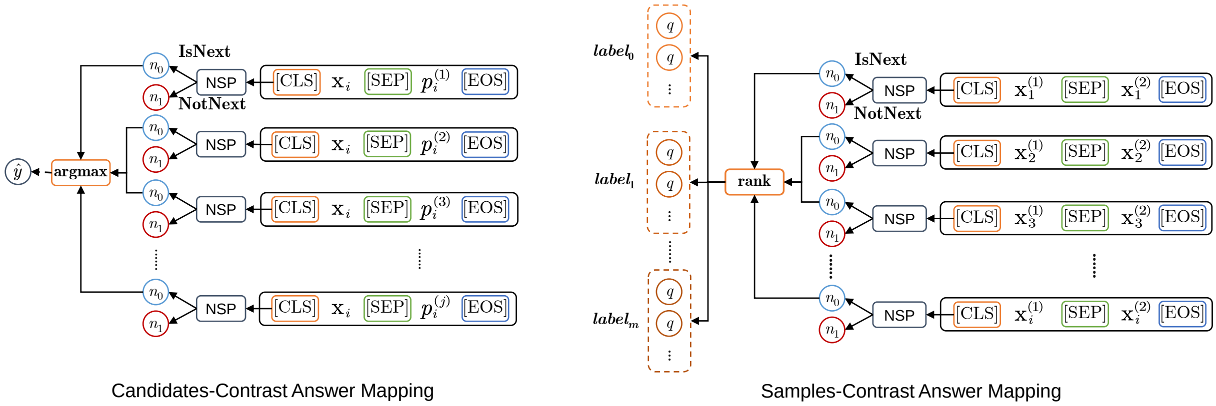

Figure 5: Two answer mapping methods candidates-contrast method (Left) and samples-contrast method (Right).

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 ← q_M(n = IsNext | x_i^(1), x_i^(2))
end for
{B_j}_{j=1}^{⌈N/bs⌉} ← divide(D, bs)
for j = 1, ..., ⌈N/bs⌉ do
    B'_j = {x_r(1), ..., x_r(bs)} ← rank(B_j, q_i, o)
    {B_m}_{m=1}^{M} ← divide(B'_j, d)
    for i = 1, ..., bs do
        ŷ_i ← m where x_i ∈ B_m
    end for
end for
```

we divide DuEL2.0 into two parts: entity linking and entity typing.

English Datasets In order to comprehensively verify the performance of NSP-BERT, we follow (Gao et al., 2021) and conduct a systematic study across 8 single-sentence and 7 sentence-pair English tasks, including 8 tasks form the GLUE benchmark (Wang et al., 2019).

4.2 Baselines

Following the FewCLUE (Xu et al., 2021) [4], we mainly choose 3 training scenarios, fine-tuning, few-shot and zero-shot.

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 softmax$(\mathbf{W}\mathbf{h}_{\text{[CLS]}})$, where $\mathbf{W} \in \mathbb{R}^{|\mathcal{Y}|\times H}$, $|\mathcal{Y}|$ is the number of labels.

Few-Shot In few-shot scenario, we choose token-level model PET (Schick and Schütze, 2021b,a) and its opitmized models ADAPET (Tam et al., 2021), P-tuning (Liu et al., 2021) and LM-BFF(Gao et al., 2021). We also choose sentence-level model EFL (Wang et al., 2021). All few-shot models are trained on FewCLUE's training set.

Zero-Shot In zero-shot scenario, there are two ways to realize, one is GPT-Zero using L2R LM (Radford et al., 2018, 2019; Brown et al., 2020), the other is PET-Zero using MLM (Schick and Schütze, 2021b,a).

4.3 Experiment Settings

For Chinese tasks in FewCLUE and DuEL2.0, we follow the settings in (Xu et al., 2021) and use RoBERTa-wwm-ext (Cui et al., 2019, 2020) [5], a Chinese RoBERTa-base model with whole-word-mask, for the baselines, which is expected to have better performance on cloze-style tasks. The GPT model is NEZHA-Gen (Wei et al., 2019) [6].

Because of the need to utilize the model pre-trained by the NSP task, none of the RoBERTa models are suitable for our NSP-BERT. So we adopt the vanilla BERT trained by UER using MLM and NSP (Zhao et al., 2019) [7]. The pre-training corpus is a large mixed corpus in Chinese. Along with the base model, we conduct experiments using UER-BERTs of various scales (tiny, small, and big) to validate the effect of NSP-BERT. Meanwhile, we use models trained by other or-

[4] https://github.com/CLUEbenchmark/FewCLUE

[5] https://github.com/ymcui/Chinese-BERT-wwm

[6] https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/NEZHA-Gen-TensorFlow

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

Method	Score	Single-Sentence EPRSTMT	CSLDCP	TNEWS	IFLYTEK	Sentence-Pair OCNLI	BUSTM	CSL	Others ChID	CLUEWSC
Human	*82.50*	*90.0*	*68.0*	*71.0*	*66.0*	*90.3*	*88.0*	*84.0*	*87.1*	*98.0*
Majority	*29.04*	*50.0*	*1.5*	*6.7*	*0.8*	*38.1*	*50.0*	*50.0*	*14.3*	*50.0*
Fine-Tuning†	42.80	63.2	35.7	49.3	32.8	33.5	55.5	50.0	15.7	49.6
PET†	57.37	87.2	**56.9**	53.7	35.1	43.9	64.0	55.0	**61.3**	**59.2**
ADAPET†	50.90	**89.0**	43.3	54.8	36.3	37.0	**69.7**	52.1	22.2	53.9
P-tuning†	**59.91**	88.3	56.0	54.2	**57.6**	41.9	60.9	62.9	59.3	58.1
LM-BFF†	55.80	84.6	53.6	**56.3**	46.1	43.1	54.1	51.2	**61.3**	51.8
EFL†	56.54	85.6	46.7	53.5	44.0	**67.5**	67.6	**61.6**	28.2	54.2
GPT-ZERO	43.40	57.5	26.2	37.0	19.0	34.4	50.0	50.1	**65.6**	50.3
PET-ZERO	45.10	85.2	12.6	26.1	26.6	**40.3**	50.6	52.2	57.6	54.7
NSP-BERT$_{\text{Ours}}$	**55.96**	**86.9**	**47.6**	**51.0**	**41.6**	37.4*	**63.4***	**64.4***	52.0	59.4*

Table 1: Main results on Chinese benchmark FewCLUE. We report the accuracy on all 9 tasks and calculate the average accuracy as the score of all tasks. †: using FewCLUE training set. Otherwise, no training samples are used. *: using of samples-contrast answer mapping method.

	Single-Sentence SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)	Sentence-Pair MNLI(m/mm) (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (Pear.)
Fine-Tuning (full)‡	*95.0*	*58.7*	*90.8*	*89.4*	*87.8*	*97.0*	*97.4*	*62.6*	*89.8 / 89.5*	*92.6*	*93.3*	*80.9*	*91.4*	*81.7*	*91.9*
Fine-Tuning (few)†	81.4	43.9	76.9	75.8	72.0	90.8	88.8	33.9	45.8 / 47.8	48.4	60.2	54.4	76.6	60.7	53.5
Majority	*50.9*	*23.1*	*50.0*	*50.0*	*50.0*	*50.0*	*18.8*	*0.0*	*32.7 / 33.0*	*33.8*	*49.5*	*52.7*	*81.2*	*0.0*	-
PET-ZERO	**83.6**	**35.0**	**80.8**	**79.5**	67.6	51.4	32.0	**2.0**	**50.8 / 51.7**	**49.5**	50.8	51.3	61.9	49.7	-3.2
NSP-BERT$_{\text{Ours}}$	78.0	33.1	75.2	76.9	**75.4**	**59.3**	**48.6**	-5.3	39.4 / 39.2	43.4	**67.6**	**55.6**	**71.4**	**59.0**	**63.9**

Table 2: Results on English datasets using BERT-LARGE. Since NSP-BERT has no obvious advantage on the English datasets, we only compared with PEF-ZERO, using RoBERTa-LARGE and the manual prompt templates following (Gao et al., 2021). ‡: full training set is used (see dataset sizes in Table 7); †: $K = 16$ (per class) for few-shot experiments. Otherwise, no training samples are used. Majority: majority class.

ganizations (Google[8] and HFL[5]), to evaluate the robustness of our optimization methods.

For English tasks, we follow the settings in (Gao et al., 2021). We use RoBERTa-LARGE[9] for PET, and vanilla English BERT-LARGE[8] for our NSP-BERT.

4.4 Main Results

The table 1 reports the main results on FewCLUE. Our NSP-BERT model outperformed all other zero-shot learning methods on 7 out of 9 datasets. Its performance is comparable to the best few-shot methods currently available. When using the same size model, it outperforms GPT-ZERO (based on L2R LM) and PET-ZERO (based on MLM) significantly on the single-sentence classification tasks (**CSLDCP, TNEWS and IFLTEK**). It demonstrates NSP's remarkable ability to distinguish across sentence topics in Chinese tasks. Nonetheless, as discussed in the previous section, the sentence-level prompt-learning methods have a number of drawbacks when used with cloze-style tasks, and NSP-BERT is no exception. This demonstrates that we have a gap in **ChID** when compared to token-level methods.

Table 2 shows the results on English datasets. Although our method does not achieve the SOTA level on most tasks, it is still competitive compared to the token-level PET model. This shows that NSP-BERT is universal in different languages.

4.5 Analysis

Two-Stage Prompt In §3.2, we introduced a two-stage prompt method for word sense disambiguation tasks. We compare its effect with a one-stage prompt on dataset DuEL2.0. Our model has satisfactory performance on DuEL2.0 without relying on any training data, especially for entity linking, NSP-BERT can handle entity descriptions of different lengths well, which is something that models such as PET can hardly achieve.

Influence of Prompt's Logic and Fluency The biggest difference between NSP-BERT and contrast learning is that the prompts in NSP-BERT need to be close to natural language habits. As shown in Figure 7, based on the 3 prompt templates (see Appendix 14), according to the logic, $T_3 > T_2 > T_1$, the accuracy increased significantly,

[8]https://github.com/google-research/bert
[9]https://huggingface.co/roberta-large

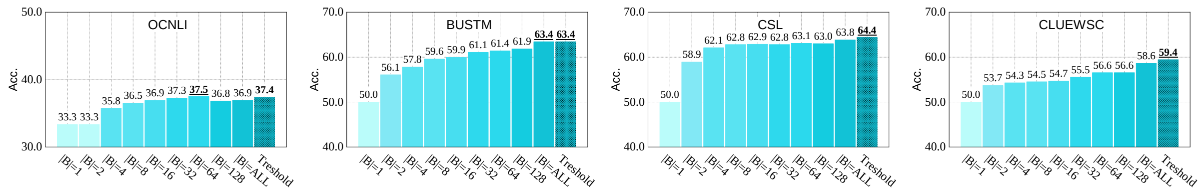

Figure 6: The performance of the samples-contrast answer mapping method under different preconditions on OCNLI, BUSTM, CSL and CLUEWSC. Batch size $|\mathcal{B}| \in \{1, 2, ..., 128, \text{ALL}\}$, when the batch size is 1 (1 and 2 for OCNLI), the result is a random guess, when the batch size is ALL, 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.

ORG	Models	Entity Linking One-S	Entity Linking Two-S	Entity Typing One-S	Entity Typing Two-S
Google[8]	BERT-Chinese	60.77	**66.99↑**	24.08	**31.18↑**
HFL[5]	BERT-wwm	57.86	**66.64↑**	23.99	**28.64↑**
	BERT-wwm-ext	59.03	**66.82↑**	24.25	**31.71↑**
UER[7]	BERT-mixed	61.16	**69.66↑**	31.35	**40.04↑**
Baselines	GPT-ZERO	/	/	28.48	
	PET-ZERO	/	/	**40.46**	

Table 3: Results (Acc.) of NSP-BERT on DuEL2.0 with one-stage prompt (One-S) and two-stage prompt (Two-S). Since GPT-ZERO and PET-ZERO are hard to handle variable length entity description, we can not report their performance on entity linking.

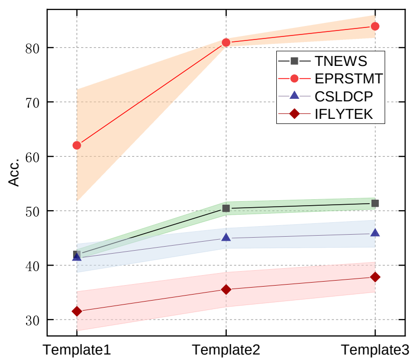

Figure 7: When prompts become more fluent and logical, the accuracy of NSP-BERT improves.

on 4 datasets (EPRSTMT, TNEWS, CSLDCP and IFLYTEK).

Samples-Contrast As shown in Table 4, if we use the same prompt method like PET, the result is close to random guessing. But when we compare the NSP output probabilities between samples, the performance improved significantly. 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.

Meanwhile, the performance of samples-contrast on sentence-pair task make us to rethink the NSP task in BERT's pre-training process. The reason

	MNLI(m/mm)	SNLI	QNLI	RTE	MRPC
Majority	*32.7 / 33.0*	*33.8*	*49.5*	*52.7*	*81.7*
PET-like	38.1 / 34.1	34.1	52.8	53.4	53.2
S-C	**39.4 / 39.2**	**43.4**	**67.6**	**55.6**	**71.4**

Table 4: PET-like: using the similar prompt method as PET; S-C: Samples-Contrast method.

that RoBERTa and others models remove NSP during pre-training, perhaps because NSP makes the output probability of most sentence pairs approach 1 (show in Appendix B.2), which makes the initialization of the model not good enough when handling sentence-pair task such as NLI and question answering[10]. This result is not only caused by NSP-head, but a large part of the main layer and segment embeddings of BERT affected by NSP.

5 Conclusion

In this paper, we introduce NSP-BERT, which uses an unexpected pre-training task Next Sentence Prediction (NSP) of BERT to perform various NLP tasks using prompts. As a sentence-level prompt-learning method, NSP-BERT not only can achieve SOTA results on multiple tasks, but it also has an impressive improvement over prior zero-shot methods (GPT and PET) in Chinese benchmark FewCLUE. NSP-BERT can accomplish non-fixed length tasks that are difficult to be solved by token-level methods, such as entity linking tasks with variable-length entity descriptions. Our NSP-BERT is inspiring for prompt-based learning owing to our experiments show that a simple pre-training task can efficiently solve various downstream tasks without any task-specific training data.

[10] These tasks need to optimize the output probability of sentence pairs to close to 0 or 1.

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.

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 *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7057–7067.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. In *ACL 2021: 59th annual meeting of the Association for Computational Linguistics*, pages 1835–1845.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

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 *the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *the Third International Workshop on Paraphrasing (IWP2005)*.

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 *Advances in Neural Information Processing Systems*, volume 32, pages 13042–13054.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL 2021: 59th annual meeting of the Association for Computational Linguistics*, pages 3816–3830.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

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 *ACM SIGKDD international conference on Knowledge discovery and data mining*.

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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3507–3512.

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 *The 2020 Conference On Empirical Methods In Natural Language Processing*.

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*, 8:64–77.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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. `https://github.com/xiaobu-coai/BUSTM`.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.

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 *In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. (pp. pp. 2463-2473). Association for Computational Linguistics: Hong Kong, China. (2019)*, pages 2463–2473.

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*, 1(8):9.

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*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Livio Baldini Soares, Nicholas Arthur FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

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

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

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*.

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.

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

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. 7.

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*, 39(2-3).

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

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1112–1122.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

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 *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772.

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 *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. In *ICLR 2021: The Ninth International Conference on Learning Representations*.

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 pretraining models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 241–246.

Chujie Zheng, Minlie Huang, and Aixin Sun. 2019. Chid: A large-scale chinese idiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 778–787.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

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 original text as $t_{\leqslant l-1}$, the prompt as $t_{l:Z}$, the label span which will be predicted as $t_{l:r}$, and it will be replaced by $\texttt{[MASK]}_{l:r}$. When ignoring special tokens such as `[CLS]`and `[PAD]`, the input of PET-ZERO is:

$$\mathbf{x}_{input} = t_1, ..., t_{l-1}, \texttt{[MASK]}_l, ..., \texttt{[MASK]}_r, t_{r+1}, ..., t_Z. \tag{6}$$

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

$$q(y_i^{(j)}|\mathbf{x}_i) = \underset{1\leqslant j\leqslant M}{\text{softmax}}\left(\prod_{l\leqslant v\leqslant r} q_{\mathcal{M}_{\text{MLM}}}(t_v^{(j)}|\mathbf{x}_{input})\right). \tag{7}$$

GPT-ZERO For Left-2-Right language model, the prompt is $t_{l:r}^{(j)}$, and tokens will input one by one, when the current token of prompt is $t_v^{(j)}$, the condition input is :

$$\mathbf{x}_{input} = t_1, ..., t_{l-1}, \texttt{[SEP]}, t_l^{(j)}, ..., t_{v-1}^{(j)}. \tag{8}$$

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

$$q(y_i^{(j)}|\mathbf{x}_i) = \underset{1\leqslant j\leqslant M}{\text{softmax}}\left(\prod_{l\leqslant v\leqslant r} q_{\mathcal{M}_{\text{L2R}}}(t_v^{(j)}|\mathbf{x}_{input})\right). \tag{9}$$

NSP-BERT For our NSP-BERT, the prompt $t_{l:r}^{(j)}$ will be inputed at once:

$$\mathbf{x}_{input} = t_1, ..., t_{l-1}, \texttt{[SEP]}, t_l^{(j)}, ..., t_r^{(j)}. \tag{10}$$

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

$$q(y_i^{(j)}|\mathbf{x}_i) = \underset{1\leqslant j\leqslant M}{\text{softmax}}(q_{\mathcal{M}_{\text{NSP}}}(\mathbf{x}_{input})). \tag{11}$$

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 [11] 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[12] for the baselines.

Table 5 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
GPT	12	768	12	102M	-
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 5: The parameters of different models used in our experiment. Denote the number of layers as L, the hidden size as H, and the number of self-attention heads as A. "-" means not used in our paper; ZH means Chinese model; EN means 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.

[11] https://github.com/princeton-nlp/LM-BFF
[12] https://github.com/pytorch/fairseq/tree/main/examples/roberta

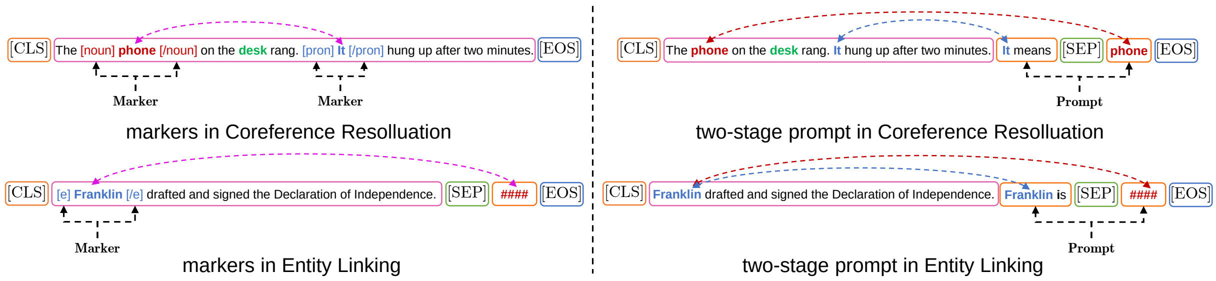

Figure 8: The comparison of markers (Left) and two-stage prompt (Right), examples in coreference resolution and entity linking/typing tasks.

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 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[13]. 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 6: 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), we evaluate our model on 8 single-sentence and 7 sentence-pair English tasks, including 8 tasks from the GLUE benchmark (Wang et al., 2019). For the datasets in GLUE, including SST-2 (Socher et al., 2013), CoLA (Warstadt et al., 2019), 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), MRPC (Dolan and Brockett, 2005), QQP [14] and STS-B (Cer et al., 2017), we follow (Gao et al., 2021) and (Zhang et al., 2021) and use their original development sets for testing. For datasets MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), MPQA (Wiebe et al., 2005), Subj (Pang and Lee, 2004), we use the testing set randomly sampled from training set and leaved from training by (Gao et al., 2021)[15]. For SNLI (Bowman et al., 2015),

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

[14]https://www.quora.com/q/quoradata/

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

Category	Corpus	|Train|	|Dev|	|Test|	$\|\mathcal{Y}\|$	Task Type	Metrics	Source
				Tasks in Chinese (FewCLUE)				
Single-Sentence	EPRSTMT	32	32	610	2	Sentiment Analysis	Acc.	E-commerce Reviews
	TNEWS	240	240	2,010	15	Short Text Classification	Acc.	News Title
	CSLDCP	536	2,068	1,784	67	Long Text Classification	Acc.	Academic CNKI
	IFLYTEK	928	690	1,749	119	Long Text Classification	Acc.	App Description
Sentence-Pair	OCNLI	32	32	2,520	3	Natural Language Inference	Acc.	5 genres
	BUSTM	32	32	1,772	2	Short Text Matching	Acc.	AI Virtual Assistant
	CSL	32	32	2,828	2	Keyword Recognition	Acc.	Academic CNKI
Others	ChID	42	42	2,002	7	Chinese Idiom Cloze Test	Acc.	Novel, Essay News
	CLUEWSC	32	32	976	2	Coreference Resolution	Acc.	Chinese Fiction Books
				Tasks in English (GLUE and more)				
Single-Sentence	SST-2	6,920	32	872	2	Sentiment Analysis	Acc.	Movie Reviews
	SST-5	8,544	80	2,210	5	Sentiment Analysis	Acc.	Movie Reviews
	MR	8,662	32	2,000	2	Sentiment Analysis	Acc.	Movie Reviews
	CR	1,775	32	2,000	2	Sentiment Analysis	Acc.	E-commerce Reviews
	MPQA	8,606	32	2,000	2	Opinion Polarity	Acc.	World Press
	Subj	8,000	32	2,000	2	Subjectivity	Acc.	Movie Reviews
	TREC	5.452	96	500	6	Question Classification	Acc.	Ad Hoc Articles
	CoLA	8,551	32	1,042	2	Acceptability	Matt.	Books and Journal Articles
Sentence-Pair	MNLI	392,702	48	9,815	3	Natural Language Inference	Acc.	Speech, Fiction and Reports
	MNLI-mm	392,702	48	9,832	3	Natural Language Inference	Acc.	Speech, Fiction and Reports
	SNLI	549,367	48	9,842	3	Natural Language Inference	Acc.	Image Captions
	QNLI	104,743	32	5,463	2	Natural Language Inference	Acc.	Wikipedia
	RTE	2,490	32	277	2	Natural Language Inference	Acc.	News and Wikipedia
	MRPC	3,668	32	408	2	Paraphrase	F1	Online News
	QQP	363,846	32	40,431	2	Paraphrase	F1	Quora Community
	STS-B	5,749	96	1,500	$\mathcal{R}$	Sentence Similarity	Pear.	News, Video and Images

Table 7: 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.

SST-5 (Socher et al., 2013) and TREC (Voorhees and Tice, 2000), we use their official test sets.

As shown in Table 7, the size of the training set and development set is determined by the number of labels, which is $K \times |\mathcal{Y}|$, and $K = 16$. Since STS-B is a real-valued regression task which ranged from 0 to 5, we treat it as an integer classification problem with label set $\{0, 1, 2, 3, 4, 5\}$, then the size of development set is 6×16.

B.2 Results

Different Model Scales In order to better show the effectiveness of NSP-BERT, we compared the impact of the models' scale on FewCLUE, shown in Figure 9. The average accuracy of tiny, small, base and large BERT models are 47.35, 49.69, 56.95 and 57.0 respectively, when the baselines GPT-Zero and PER-Zero are 43.40 and 45.10.

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.

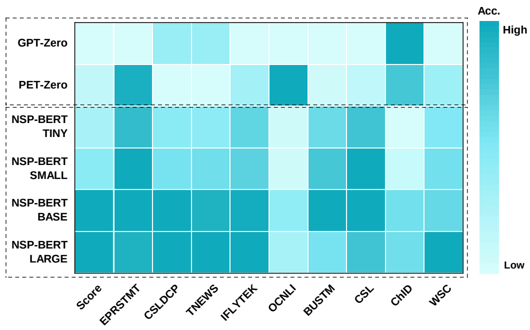

Figure 9: Sketch of accuracy for different scales of models. X-axis represents the tasks in FewCLUE and the y-axis represents the baselines (GPT-Zero and PET-Zero) and NSP-BERT at different model scales (tiny, small, base and large).

- **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 8, Table 9, Table 10 and Table 11.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
Goo.	BERT-Chinese	70.63/72.30	75.63/79.84	**76.88/83.11**
HFL	BERT-wwm	68.13/69.34	72.50/81.48	**76.25/81.97**
	BERT-wwm-ext	53.75/51.80	75.00/81.31	**81.88/83.61**
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 8: Accuracy of NSP-BERT on EPRSTMT.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
Goo.	BERT-Chinese	45.00/43.18	48.91/51.39	**51.73/52.38**
HFL	BERT-wwm	44.63/41.79	**51.00/50.75**	49.09/50.05
	BERT-wwm-ext	45.72/41.14	52.09/50.90	**52.10/51.94**
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 9: Accuracy of NSP-BERT on TNEWS.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
Goo.	BERT-Chinese	40.03/40.36	43.96/45.12	**43.96/46.02**
HFL	BERT-wwm	42.89/45.07	44.92/46.52	**45.60/47.31**
	BERT-wwm-ext	38.10/39.18	40.18/**42.32**	**41.30**/42.21
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 10: Accuracy of NSP-BERT on CSLDCP.

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 10 and Figure 11. 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.

ORG	Models	Template 1 (Dev/Test)	Template 2 (Dev/Test)	Template 3 (Dev/Test)
Goo.	BERT-Chinese	31.97/31.33	39.18/34.53	**41.59/37.56**
HFL	BERT-wwm	31.25/29.96	38.02/34.19	**40.64/37.05**
	BERT-wwm-ext	29.86/28.30	36.20/33.16	**39.83/35.05**
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 11: Accuracy of NSP-BERT on IFLYTEK.

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 $|B| \in \{1, 2, ..., 128, \mathrm{ALL}\}$, to observe the results predicted by samples-contrast method (see Table 12). 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 13. 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. The samples-contrast method is necessary for the sentence-pair tasks.

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 14, and briefly report the prompts of English datasets in Table 15.

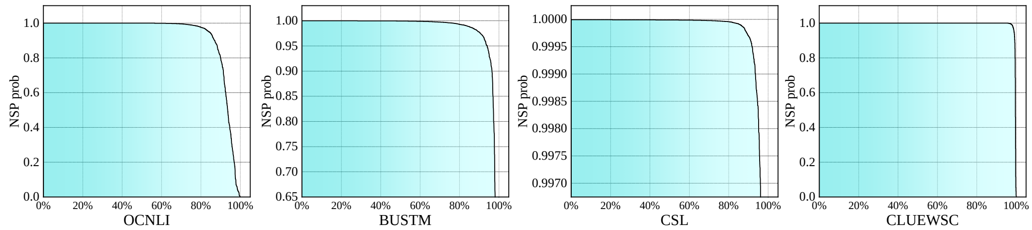

Figure 10: The NSP output probability of the 4 sentence-pair tasks OCNLI, BUSTM, CSL and CLUEWSC in Chinese benchmark FewCLUE. The x-axis represents the proportion of the samples. And the y-axis represents the NSP probability of the samples.

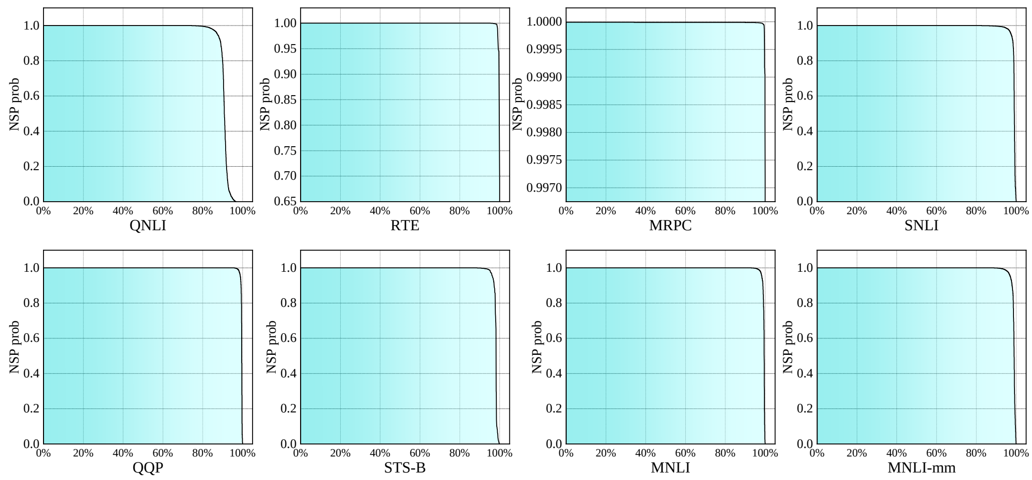

Figure 11: The NSP output probability of the 8 English sentence-pair tasks QNLI, RTE, MRPC, SNLI, QQP, STS-B, MNLI and MNLI-mm. The x-axis represents the proportion of the samples. And the y-axis represents the NSP probability of the samples.

Dataset	**Dev**	**Test**									
		$\|\mathcal{B}\|$ =1	$\|\mathcal{B}\|$ =2	$\|\mathcal{B}\|$ =4	$\|\mathcal{B}\|$ =8	$\|\mathcal{B}\|$ =16	$\|\mathcal{B}\|$ =32	$\|\mathcal{B}\|$ =64	$\|\mathcal{B}\|$ =128	$\|\mathcal{B}\|$ =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
MRPC	50.00	79.87	61.19	62.19	63.28	63.48	63.88	63.58	63.18	63.18	**71.38**
QQP	75.00	53.82	52.75	54.36	55.57	56.18	56.46	56.64	56.70	56.77	**58.97**
STS-B	57.28	-	-	-	50.59	54.94	57.25	59.39	61.62	**66.24**	63.92

Table 12: 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 Mapping	C-C	✓	✓	✓	✓				✓		✓	✓
	S-C					✓	✓	✓		✓		

Table 13: 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	Prompt Templates	Label Names
EPRSTMT	**Template 1**: The screen stopped working. `[SEP]` `[label]`. **Template 2**: The screen stopped working. `[SEP]` I am `[label]`. **Template 3**: The screen stopped working. `[SEP]` I am very `[label]` about this shopping.	**2 labels**: Positive (Happy); Negative (Sad)
TNEWS	**Template 1**: La Liga: Atletico Madrid VS Espanyol. `[SEP]` `[label]`. **Template 2**: La Liga: Atletico Madrid VS Espanyol. `[SEP]` `[label]` news. **Template 3**: La Liga: Atletico Madrid VS Espanyol. `[SEP]` This is a piece of `[label]` news.	**15 labels**: Education; Finance; House; Travel; Technology; Sports; Game; Culture; Car; Story; Entertainment; Military; Agriculture; World; Stock.
CSLDCP	**Template 1**: Grove Mountains (GRV) 020043 is a special chondrite.... `[SEP]` `[label]`. **Template 2**: Grove Mountains (GRV) 020043 is a special chondrite.... `[SEP]` `[label]` paper. **Template 3**: Grove Mountains (GRV) 020043 is a special chondrite.... `[SEP]` This is a paper about `[label]`.	**67 labels**: Materials Science and Engineering; Crop Science; Stomatology; Pharmacy; Pedagogy; Water Conserv-ancy Engineering; Theoretical Economics; Food Science and Engineering; Animal Science/Veterinary Science ; ...
IFLYTEK	**Template 1**: GooglePlay is Google's official application market... `[SEP]` `[label]`. **Template 2**: GooglePlay is Google's official application market... `[SEP]` `[label]` app. **Template 3**: GooglePlay is Google's official application market... `[SEP]` It's a `[label]` app.	**119 labels**: Taxi; Map Navigation; Free WIFI; Car Rental; Same City Service; Express Logistics; Wedding; House-keeping; Public Transportation; Government Affairs; Community Services; Fleece; Magic; Xian Xia; Card; Flying Air Combat; Shooting Game; Leisure Puz; ...
OCNLI	The two people came back from Japan the day before yesterday. `[SEP]` The two of them stayed in Japan for a week.	**3 labels**: Contradiction; Neutral; Entailment.
BUSTM	Sing me a song. `[SEP]` Play a song for us.	**2 labels**: Matched; Unmatched.
ChID	This means that in the near future, HJT heterojunction cells may usher in an explosion, and photovoltaic cells may also usher in a `[BLANK]` opportunity period from PERC to HJT. `[SEP]` historically revolutionary.	**7 candidates** (Each sample has different candidates): stand ready; historically revolutionary; absolutely irreconcilable; far away; return to the original owner; waves and clouds; strut.
CLUEWSC	The phone on the desk rang. It hung up after two minutes. It means `[SEP]` phone.	**2 labels**: True; False.
DuEL2.0 Entity Linking	Franklin drafted the Declaration of Independence. Franklin is `[SEP]` he is the founding Fathers of the United States...	**5.37 entities per sample**: **Entity 1**: The founding Fathers of the United States. American politician, physicist and social activist. **Entity 2**: American female swimmer, good at short backstroke and freestyle, nicknamed "female flying fish". **Entity 3**: British captain and Arctic explorer, served on the Bellerophon in the early years and participated in the Battle of Trafalgar.
DuEL2.0 Entity Typing	Franklin drafted the Declaration of Independence. Franklin is `[SEP]` he is a person...	**24 types**: Event; Person; Work; Location; Time and Calendar; Brand; Natural and Geography; Game; Biological; Medicine; Food; Software; Vehicle; Website; Disease and Symptom; Organization; Awards; Education; Culture; Constellation; Law and Regulation; Virtual-Things; Diagnosis and Treatment; Other.

Table 14: The prompts used for tasks in FewCLUE. `[label]` is the token will be replaced by the mapping words.. Since there are two options for the prompt, **prefix** and **suffix**, we select the most suitable one through the development set. **The original datasets are all in Chinese**, in order to facilitate understanding, we have performed a certain conversion. Especially for the ChID dataset, since idioms are a relatively specific linguistic phenomenon in Chinese, most idioms are composed of 4 tokens, so we only use the general cloze-sytle task to show its Prompt. For dataset with a lot of labels, due to space considerations, we have omitted some of them. The underlined part is the prompt template, otherwise it is the original text.

Task	Prompt Templates
SST-2	Original Labels: negative; positive Mapping Words: terrible; great Prompt Template: That is `[label]`. `[SEP]` $\mathbf{x}$
SST-5	Original Labels: very negative; negative; neutral; positive; very positive Mapping Words: terrible; bad; okay; good; great Prompt Template: $\mathbf{x}$ `[SEP]` That is `[label]`.
MR	Original Labels: negative; positive Mapping Words: terrible; great Prompt Template: A `[label]` piece of work. `[SEP]` $\mathbf{x}$
CR	Original Labels: positive; negative Mapping Words: terrible; great Prompt Template: A `[label]` piece of work. `[SEP]` $\mathbf{x}$
MPQA	Original Labels: positive; negative Mapping Words: terrible; great Prompt Template: A `[label]` piece of work. `[SEP]` $\mathbf{x}$
Subj	Original Labels: subjective; objective Mapping Words: exciting; normal Prompt Template: A `[label]` piece of work. `[SEP]` $\mathbf{x}$
TREC	Original Labels: description; entity; abbreviation; human; location; numeric Mapping Words: definition; entity; abbreviations; people; place; number Prompt Template: The answer is about a `[label]`. `[SEP]` $\mathbf{x}$
CoLA	Original Labels: not_grammatical; grammatical Mapping Words: wrong; correct Prompt Template: The grammar of this sentence is `[label]`. `[SEP]` $\mathbf{x}$
MNLI-m/mm	Original Labels: contradiction; neutral; entailment Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$
SNLI	Original Labels: contradiction; neutral; entailment Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$
QNLI	Original Labels: not_entailment; entailment Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$
RTE	Original Labels: not_entailment; entailment Prompt Template: $\mathbf{x}^{(2)}$ `[SEP]` $\mathbf{x}^{(1)}$
MRPC	Original Labels: not_equivalent; equivalent Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$
QQP	Original Labels: not_equivalent; equivalent Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$
STS-B	Original Labels: $[0, 5]$ Mapping Integers: 0, 1, 2, 3, 4, 5 Prompt Template: $\mathbf{x}^{(1)}$ which means. `[SEP]` $\mathbf{x}^{(2)}$

Table 15: The prompts used in English datasets. We only show the template with best performance. `[label]` is the token will be replaced by the mapping words.