applied sciences

MDPI

Article

VPN: Variation on Prompt Tuning for Named-Entity Recognition

Niu Hu [1,†], Xuan Zhou [2,†], Bing Xu [3], Hanqing Liu [1], Xiangjin Xie [1] and Hai-Tao Zheng [1,4,*]

[1] Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; hn20@mails.tsinghua.edu.cn (N.H.)
[2] PAII Inc., Palo Alto, CA 94306, USA
[3] Ping An Technology (Shenzhen) Co., Ltd., Shenzhen 518063, China
[4] Pengcheng Laboratory, Shenzhen 518000, China
* Correspondence: zheng.haitao@sz.tsinghua.edu.cn
† These authors contributed equally to this work.

Abstract: Recently, prompt-based methods have achieved a promising performance in many natural language processing benchmarks. Despite success in sentence-level classification tasks, prompt-based methods work poorly in token-level tasks, such as named entity recognition (NER), due to the sophisticated design of entity-related templates. Note that the nature of prompt tuning makes full use of the parameters of the mask language model (MLM) head, while previous methods solely utilized the last hidden layer of language models (LMs) and the power of the MLM head is overlooked. In this work, we discovered the characteristics of semantic feature changes in samples after being processed using MLMs. Based on this characteristic, we designed a prompt-tuning variant for NER tasks. We let the pre-trained model predict the label words derived from the training dataset at each position and fed the generated logits (non-normalized probability) to the CRF layer. We evaluated our method on three popular datasets, and the experiments showed that our proposed method outperforms the state-of-the-art model in all three Chinese datasets.

Keywords: prompt tuning; MLM head; NER

Citation: Hu, N.; Zhou, X.; Xu, B.; Liu, H.; Xie, X.; Zheng, H.-T. VPN: Variation on Prompt Tuning for Named-Entity Recognition. *Appl. Sci.* **2023**, *13*, 8359. https://doi.org/10.3390/app13148359

Academic Editors: Rui Araújo and Vincent A. Cicirello

Received: 23 March 2023
Revised: 10 May 2023
Accepted: 17 May 2023
Published: 19 July 2023

1. Introduction

Recently, pre-trained language models (LMs), such as BERT [1] and RoBERTa [2], have achieved a dominant performance on almost all natural language processing (NLP) tasks, upon simply fine-tuning these LMs with an extra task-specific head with task-specific training data in the downstream tasks. Despite the effectiveness and simplicity of fine-tuning LMs, there is still a wide gap between the objective functions of the pre-training and fine-tuning phases. A common conclusion in the literature [3,4] is that this mismatch results in the under-utilization of these powerful LMs.

Prompt-based approaches [5–9] have been proposed to address this problem. Unlike traditional supervised learning, which solely utilizes the parameters in LMs with rich distributed knowledge, prompt-based methods reformulate a downstream task's objective forms as those in the pre-training phase, directly modelling the probability of words without using any task-specific layers [3]. As shown in Figure 1, the sentiment classification, for example, can identify the sentiment $y \in \mathcal{Y}$ towards a given input sentence $X \in \mathcal{D}$. In traditional LM fine-tuning, we take the softmax of the special word, such as `[CLS]`, and the true label y as the loss function to further train the LM. Then, we obtain the predicted label $\hat{y}$ as the sentiment predicted. In typical prompt tuning, we add a template $T = [e_1, e_2, ..., \texttt{[MASK]}, ..., e_t]$ containing a `[MASK]` special token to the original input sequence X, then feed the new sequence $X' = [X, T]$ into the LM and let the LM predict the `[MASK]` token of the target token in the vocabulary, indicating the sentiment of the original input. Recent efforts show that prompt-based methods, as shown above, have achieved promising results in many sentence-level NLP tasks, such as natural language inference [10], sentence

classification [5], and factual probing [11]. Despite success in sentence-level classification tasks, prompt-based methods work poorly in token-level classification tasks, such as named entity recognition (NER) and parts of speech (POS).

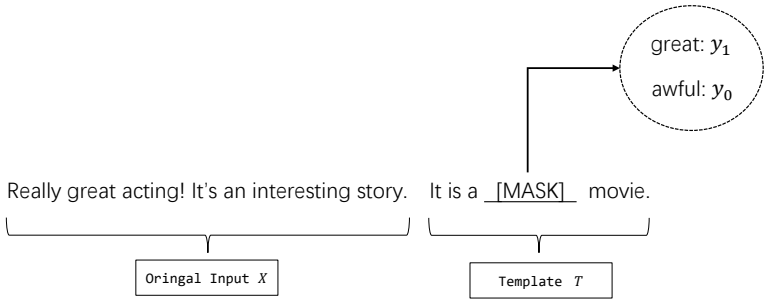

Figure 1. Prompt-based method for sentiment classification.

As a fundamental task, NER is irreplaceable in many downstream NPL tasks, such as event recognition, entity linking, etc. NER aims to put the named entity mentioned in a sentence into some pre-defined categories, such as location, person, organization, etc. Former efforts have often required an extra label-specific output dense layer, which is randomly initialized. This makes it difficult for the model to fit into an optimal point. Liu et al. [12] adopted NER prompt tuning, not introducing any extra parameters other than the parameters of the pre-trained model. They enumerated all possible entity spans and filled them in templates, meaning that inferring a sentence required feeding that sentence into the model many times. Despite its effectiveness, the enumeration procedure is time-consuming and intolerable.

Ma et al. [13] proposed a template-free prompt-tuning model for few-shot NER. They eliminated the use of templates and let the model predict class-related pivot words derived from unlabelled data instead of original words at each entity position while still predicting the original words at non-entity positions. In this way, inferring a sentence only needs the sentence to be fed into the model once. Their model gained a lot in few-shot settings while working ordinarily in rich-resource settings.

In this study, we propose a simple yet effective variation on the prompt tuning for NER. In the `BIO` scheme, the tags `B` and `I` denote that the current word is at the beginning or inside of an entity, respectively, and `O` denotes that the current word is not a component of the entity. In the `IO` scheme, the beginning of an entity is also tagged with `I`. For example, unlike Ma et al. [13], the `IO` scheme can be used to find label words; however, this makes it difficult for the model to separate several consecutive homogeneous entities, and the correlations between tags are neglected. Furthermore, the beginning and interior of an entity often convey different semantic information. For instance, the word *City* in the `LOC` entity *New York City* is more likely to be predicted as `I-LOC` rather than `B-LOC` in the `BIO` scheme, while in the `IO` scheme, the implicit semantic gaps between all three words are neglected, and all three words in *New York City* are treated equivalently. We derive the top-K tag-wise label words in the `BIO` scheme according to the frequency of occurrence and the corresponding normalized frequency. Then we let the pre-trained model predict the label words at each position and feed the generated logits (non-normalized probability) to a CRF layer to capture the correlations between the tags. We do not introduce any extra parameters other than the parameters of the pre-trained model to obtain the logits of all tags at each position.

Our contributions are as follows: (i) We found that the feature changes after the MLMs were limited, which can improve the effectiveness of the NER task and avoid introducing additional parameters. (ii) We proposed a simple yet effective variation on the prompt tuning for NER. (iii) We do not introduce any extra parameters other than the parameters of the pre-trained model to obtain the logits of all tags at each position.

(iv) Experiments show that our proposed method outperforms the state-of-the-art model on three popular datasets.

2. Related Works

In this section, we briefly introduce studies related to prompt-tuning methods and prompt tuning for NER.

2.1. Prompt Tuning

As shown above, prompt-based methods reformulate the objectives of the fine-tuning phase as a close-style objective. In this way, the gaps between the objectives of the pre-training and fine-tuning phases are bridged. GPT-3 [14] uses hand-crafted prompts for tuning and achieves a very impressive performance on various tasks, especially for few-shot learning settings. Inspired by GPT-3, many attempts [15–18] concerning knowledge probing use hand-crafted prompts to boost the models and have been widely used in relation to classification tasks [4], entailment classification and natural language inference [3,5]. Automatically generating label words and templates [19,20] avoid labour-intensive prompt design. Recently, some continuous prompts [8,21] have been proposed using learnable continuous label words and templates rather than discrete words in the vocabularies of pre-trained models.

2.2. Prompt Tuning for NER

NER is a token-level classification task that is difficult for prompt tuning. According to a popular survey of prompt tuning [3], the template design is complex for NER. To obtain templates, NER needs to enumerate all possible entity spans and types, then feed the spans and types to a pre-defined template, which is time-consuming and labour-intensive. The decoding speed increases significantly when the input sequence increases [12]. Furthermore, Ma et al. [13] proposed a one-pass decoding strategy for NER, discarding the complex template design and letting the LM predict the class-related pivot word (or label word) at the entity position. On the other hand, they claim that they did not introduce any extra parameters except for the parameters of the pre-trained model. However, they introduced extra biases when adding the special tokens corresponding to the labels in the vocabulary of the pre-trained model and set them to 0. Thus, the original biases in the pre-trained model parameters are lost.

3. Materials and Methods

3.1. Problem Setup

Given an NER dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, for each sample, $(\boldsymbol{x_i}, \boldsymbol{y_i})$ contains a word sequence $\boldsymbol{x_i} = [x_{i,t}]_{t=1}^{T}$ and its corresponding label sequence $\boldsymbol{y_i} = [y_{i,t}]_{t=1}^{T}$, where T denotes the sequence length and $y_{i,t} \in \mathcal{Y}$ is the entity type from a pre-defined entity type set $\mathcal{Y}$. The NER task aims to predict the entity type of the input word sequences in the test dataset $\mathcal{D}_{test}$ split from $\mathcal{D}$.

3.2. Label-Word Selection

As shown in Figure 2, assume we have m kinds of tags $\mathcal{Y} = \{l_j\}_{j=1}^{m}$ in dataset $\mathcal{D}$. For each tag l_j, we find all words with the label l_j from the training samples, then we select the K most frequent words $[c_{j,k}]_{k=1}^{K}$ as a representative of the label l_j.

For each representative word $c_{j,k}$, its normalized frequency is denoted as $w_{j,k}$, and the corresponding word index in the vocabulary is denoted as $d_{j,k}$. It should be noted that $d_{j,k}$ is related to the specific pre-trained model.

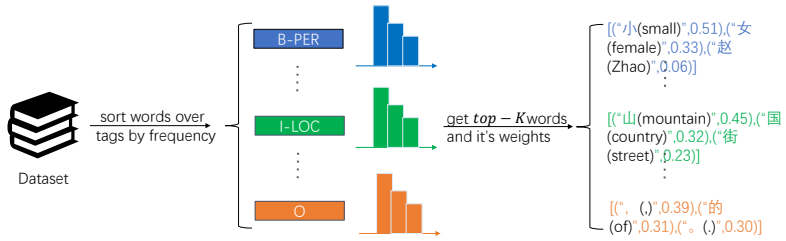

Figure 2. Label-word selection process.

In the implementation, we notice that although an entity word will not appear in both entity categories, the characters in one entity word may appear in both entity categories in the Chinese datasets. This means different entity tags in the `BIO` scheme might have the same label word. For example, character {"美"} (beautiful) with the tag `B-GPE` occurs in the `GPE` entity word {"美国"} (America), and it also occurs in the `ORG` entity word {"国美电器"} (a housekeeping appliance market) with the tag `I-GPE`. Considering that our model relies heavily on the quality of label-word selection, this co-occurrence confuses our model when distinguishing which entity type it belongs to for each word-containing character {"美"} (beautiful) . To solve this issue, we designed Algorithm 1. If the same label word occurs in different tags, we assign that label word to the tag with the maximum number of occurrences in the dataset. We first collect all the characters and sort them by the number of occurrences for each tag. Then we introduce a hyper-parameter threshold *thr* and sample $thr * K$ label words and their occurrences for each tag. The sampled results are a tag-pair dict. This hyper-parameter threshold *thr* is to ensure there are K label words for each tag in the final filtered label-word dict. Next, we merge all word-occurrence pairs together. In this word-occurrence pair list, we keep the pair with the highest occurrence and discard the rest across all pairs for a unique word. Then, for each tag, we enumerate the tag-pair dict, keep the pair that occurs in the pair list, and discard the rest. Finally, we select the top K pairs for each tag. This tag-pair dict is our final filtered label-word dict.

Algorithm 1: Label-Word Selection and Filtration

Data: Dataset $\mathcal{D}$; number of label words K; hyper-parameter *thr*
Result: Top K tag pairs dict Tag_pairs_dict'
for $word, tag$ *in* $\mathcal{D}$ **do**
  count the $(word, word\_num)$ with respect to tag
end
$Tag\_pairs\_dict \leftarrow$ **sorts** the $(word, word\_num)$ pair with respect to tag and **select** the top $K * thr$ pair
$P \leftarrow$ merges the pairs of all tags in Tag_pairs_dict
for $(word, word\_num)$ *in* P **do**
  For a unique word, **keep** the pair with the largest $word\_num$ across all pairs and **discard** the rest
end
$P' \leftarrow$ processed P
Generate the Tag_pairs_dict' over the tags for a pair in Tag_pairs_dict & P'

3.3. Dataset

The following real-world datasets are considered in our study. Table 1 shows the statistics of the datasets.

Table 1. Details of the three datasets. #ENT: number of entities; S: number of sentences; T: number of tokens.

Dataset	#ENT	Type	Train	Dev	Test
Weiobo NER	4	S	1.4k	0.3k	0.3k
		T	73.5k	14.4k	14.8k
MSRA	3	S	46.4k	-	4.4k
		T	979.2k	-	172.6k
OntoNotes 4.0	4	S	15.7k	4.3k	4.3k
		T	491.9k	200.5k	208.1k

- **OntoNotes 4.0** [22] is an annotated multilingual corpus consisting of texts from a wide variety of sources, such as telephone conversations, broadcasts, and newswires. For our NER experiment, we considered a Chinese dataset derived from OntoNotes 4.0 and processed it according to [23].
- **MSRA** [24] is a Chinese NER dataset launched in 2016 in the news domain labelled by Microsoft Research Asia. The dataset contains more than 50,000 Chinese entity identification and labelling data points. The entity category is divided into three categories: person, place, and institution.
- **Weibo NER** [25] was released in 2014 and was generated by filtering the historical data of Sina Weibo from November 2013 to December 2014. It contains 1890 Weibo messages and is labelled based on the labelled standard of the DEFT ERE of LDC2014. The dataset includes four entity categories: location, person, organization, and geopolitical entities. It includes 1350 training sets, 270 verification sets, and 270 test sets.

3.4. Implementation Details

For all our experiments, we used the `bert-base-chinese` (https://github.com/google-research/bert, accessed on 20 March 2023) pre-trained model as our backbone structure. The hidden size and number of layers of the backbone model are 768 and 12, respectively. We implemented experiments in the `TensorFlow` framework. The batch size was 8 across all our experiments. In addition, the learning rate of the CRF layer was 1×10^{-3}, and the learning rate of all other layers was 1×10^{-5}, using the AdamW optimizer with a 0.1 warm-up ratio. For small datasets, such as Weibo, we set the total epochs to 50. For MSRA and OntoNotes 4.0, we set the total epochs to 20. For evaluation, we used the `BIO` scheme. Tags `B` and `I` denote that the current word is at the beginning or inside the entity, respectively. Tag `O` denotes that the current word is not an entity component.

4. Modelling VPN

We let the LM predict several label words in the vocabulary and obtain the overall tag-related logits. These label words are more relevant to tags rather than classes. In this way, we can also model the logits of positions labelled `O` and use the `BIO` scheme rather than the `IO` scheme, which can use the CRF layer to boost the model's performance.

In this work, we consider an NER task as a sequence-to-sequence task. Figure 3 shows the overall architecture of our proposed model. Given an input sequence $\boldsymbol{x} = \{x_1, x_2, ..., x_T\}$ and the corresponding label sequence $\boldsymbol{y} = \{y_1, y_2, ..., y_T\}$, we embed each word using a pre-trained LM to obtain an embedded sequence $E_{emb} \in \mathbb{R}^{T \times d_H}$:

$$\begin{aligned} E_{emb} &= \texttt{Encoder}(\boldsymbol{x}) \\ &= [\boldsymbol{e}(x_1), \boldsymbol{e}(x_2), ..., \boldsymbol{e}(x_T)], \end{aligned} \tag{1}$$

where $\boldsymbol{e}(x_t) \in \mathbb{R}^{d_H}$ is the last layer of the hidden state of word x_t, and T and d_H denote the sequence length and hidden dimension of the transformer model, respectively.

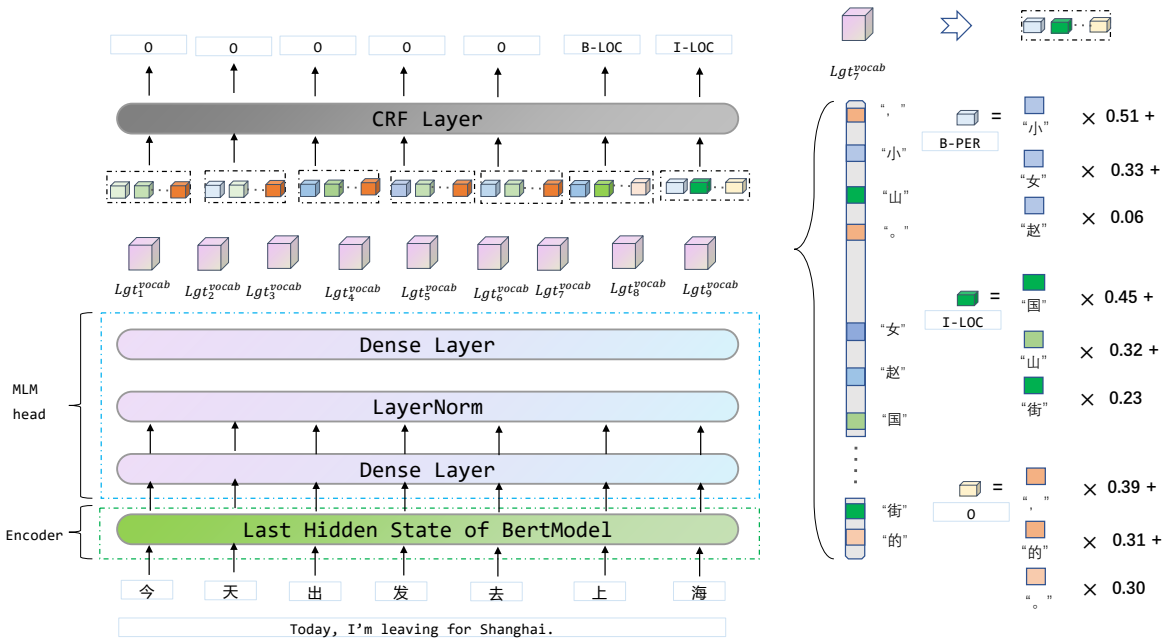

Figure 3. Architecture of our proposed model. The left side illustrates the overall process of inferring a specific sentence, while the right side illustrates the calculation of the $logit^{label}$. Chinese label words are the same as those in Figure 2.

In order to take full advantage of the pre-trained model, along with BERT's pre-training stage, we calculate the word prediction logits using the masked language model head as follows:

$$\begin{aligned} E_1 &= \texttt{Dense}_1(E_{emb}), \\ E_2 &= \texttt{LayerNorm}(E_1), \\ logit^{vocab} &= \texttt{Dense}_2(E_2), \end{aligned} \tag{2}$$

where $\texttt{Dense}_1 \in \mathbb{R}^{T \times d_H}$, $\texttt{Dense}_2 \in \mathbb{R}^{d_H \times |\mathcal{V}|}$, $logit^{vocab} \in \mathbb{R}^{T \times |\mathcal{V}|}$, and $|\mathcal{V}|$ represents the cardinal number of the vocabulary.

For each word x_t, we obtain the label logit through mean pooling corresponding to the top K representative words of the entity tags, that is,

$$logit^{label}_{t,j} = \sum_{k=1}^{K} w_{j,k} logit^{vocab}_{t,d_{j,k}}, \tag{3}$$

where $logit^{label} \in \mathbb{R}^{T \times m}$.

Then, we feed the $logit^{label}$ to the conditional random field (CRF) [26] layer. Implementation-wise, CRF computes an energy given a candidate output $\boldsymbol{y}$ and a context $\boldsymbol{x}$ (i.e., input sequence), followed by a softmax operator to obtain the conditional likelihood, i.e.,

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}) = \frac{e^{s(\boldsymbol{x},\boldsymbol{y})}}{\sum_{\tilde{\mathbf{y}} \in Y_{\text{all}}} e^{s(\boldsymbol{x},\tilde{\mathbf{y}})}}, \tag{4}$$

$$s(\boldsymbol{x},\boldsymbol{y}) = \sum_{t=1}^{T-1} logit^{label}_{t,\boldsymbol{y}_t}(\boldsymbol{x}) + A_{\boldsymbol{y}_t,\boldsymbol{y}_{t+1}}. \tag{5}$$

Here, Y_{all} is the set of all possible tag sequences, and the transition matrix $A \in \mathbb{R}^{m \times m}$ characterizes the smoothness of the label sequence (probability of switching between consequent labels).

5. Results

5.1. Baselines

In this work, we evaluated our proposed model and compared it with several competitive baselines. These baseline models include the BERT model and other BERT-related models as the backbone model. The baseline models include:

- **BERT-tagger.** BERT-tagger [1] is a strong baseline in token-level classification tasks such as NER;
- **BERT+Glyce.** Meng et al. [27] took advantage of glyph information to enrich the pictographic evidence in characters using historical Chinese scripts;
- **BERT+FLAT.** Li et al. [28] converted the character–word lattice structure into a flat structure of spans;
- **BERT-MRC.** Li et al. [29] reformulated NER as a machine reading-comprehension task.

5.2. Experimental Results

Table 2 shows our main test's F1 results. From the results, we first find that our model significantly outperforms all the baseline models, including the state-of-the-art models, on the three Chinese datasets. We owe these across-the-board gains to the reuse of the MLM head derived from the original pre-trained model, eliminating the need to design a label-specific output layer; the CRF layer is also helpful. For a small dataset, such as Weibo, compared to the vanilla BERT-tagger, the rest of the baseline models showed little improvement, while our model improved by 4–5%. For the large dataset, OntoNotes 4.0, all three baseline models improved by 3–4% compared to the vanilla BERT-tagger, while our model achieved an improvement of 4.89% compared to the BERT-tagger. For the larger dataset, MSRA, all the models achieved satisfying results, while our model marginally outperformed the baselines. From the results, we find that the size of the dataset has a huge impact on the results. Another observation in terms of small datasets, such as Weibo NER, is that the performance gains were greater in small datasets than large datasets compared to the baselines. The baseline models all have class-related output layers in which the parameters are randomly generated; this might be why they were harder to fit in a smaller dataset.

Table 2. Overall results for VPN on the three datasets.

Method	Datasets								
	Weibo			OntoNotes 4.0			MSRA		
	P	R	F1	P	R	F1	P	R	F1
BERT-tagger	67.12	66.88	67.33	78.01	80.35	79.16	94.97	94.62	94.80
BERT+Glyce	67.12	66.88	67.60	81.87	81.40	80.62	95.57	95.51	95.54
BERT+FLAT	-	-	68.55	-	-	81.82	-	-	96.09
BERT-MRC	-	-	-	82.98	81.25	82.11	96.18	95.12	95.75
VPN (our method)	78.81	73.68	73.25	80.34	84.71	82.47	96.48	95.88	96.18

5.3. Ablation Study

In Figure 4, we compare how varying the size of the candidate hyper-parameter K affects the performance. The performance peaks at a moderate K; after this the gain tapers off. This is because when using an excessive or lesser amount of K, the label words of each tag introduce some helpful or less helpful words, affecting the performance of the model. Furthermore, we can find that different datasets have different most-appropriate K values, indicating that the distribution of the data also has a great impact.

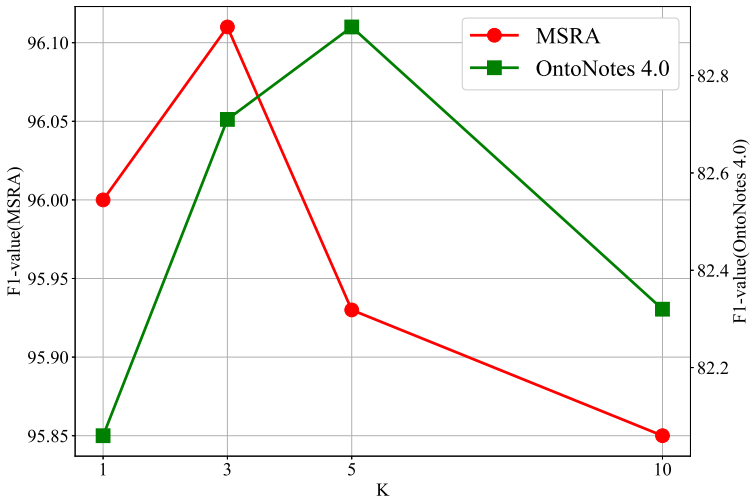

Figure 4. Performance with different candidate size *K*.

6. Discussion

6.1. Motivation of Our Method

When using prompt-based methods to solve sentence-classification tasks, researchers add a template with a special `[MASK]` token to the original input text and let the pre-trained model predict a `[MASK]` set of label words, each representing a specific pre-defined class of the original input text. In this way, the prediction ability of the `[MASK]` token is fully exploited. Intuitively, we wondered whether the non-mask token has the ability to predict. We conduct a sentence-restoration experiment to test our hypothesis. We fed the original input text to a pre-trained model and obtained the last hidden states of each token. Subsequently, we further fed the hidden states into the masked language head of a pre-trained model used in the pre-training phase and obtained the logits over the pre-trained model's vocabulary.

Here, we report the accuracy at the token level of each token, then output the tokens with the largest logit to restore the original input text. For example, we input the sentence, "今天出发去上海" (Today, I'm leaving for Shanghai) and want the model to output the original sentence. There are often tens of thousands of tokens in the pre-trained model's vocabulary, and to restore the plain non-mask original token is not easy. We conducted our experiments on two datasets: AGNews [30] and The People's Daily. Table 3 shows the results. We notice that, in English datasets, such as AGNews, the accuracy at the token level is about 0.87, while in Chinese datasets, such as The People's Daily, the accuracy is about 0.95, showing that these non-mask tokens also have the ability to predict. Note that the MLM heads in the pre-trained model can achieve remarkable results in predicting the input token without any fine-tuning. Therefore, we can let the pre-trained model's MLM head predict other label words in the vocabulary of the pre-trained model instead. In token-classification tasks, such as NER, distinguishing different token categories is required; therefore, we assign different token categories (i.e., entity tag type, e.g., `B-LOC`) with different label words and let the pre-trained model's MLM head predict the tag-related label words for each token. Summing the predicted logits for a label word of a specific token category means the logit for that token can be predicted as that token category.

Table 3. Results of the sentence-restoration experiment.

Dataset	Train	Dev
AGNews	0.87	0.86
The People's Daily	0.94	0.94

Similar to the procedure of letting the mask token predict the pre-defined label words in a sentence-classification task, we let the non-mask token predict a set of tokens as label words to solve the named-entity task. Figure 5 shows the correlation of our model in a named-entity task and prompt tuning in a classification task.

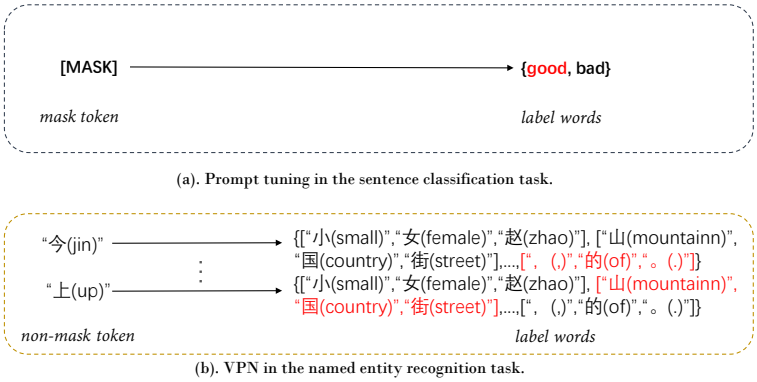

Figure 5. Correlation of our model in a named-entity task and prompt tuning in a classification task.

6.2. English Results and Future Work

Our model can be applied not only to the Chinese language, but also to other languages. We conducted a series of experiments on the English datasets CoNLL 2003 [31] and OntoNotes 5.0 [32]. CoNLL-2003 is a named-entity-recognition dataset released as a part of the CoNLL-2003 shared task language-independent named-entity recognition. The data consist of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file, and a large file with unannotated data. The English data were taken from the Reuters Corpus. This corpus consists of Reuters news stories between August 1996 and August 1997. For the training and development set, ten days' worth of data were taken from the files representing the end of August 1996. For the test set, the texts were from December 1996. The pre-processed raw data cover the month of September 1996. OntoNotes 5.0 is a large corpus comprising various genres of text (news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows) in three languages (English, Chinese, and Arabic) with structural information (syntax and predicate argument structure) and shallow semantics (word sense linked to an ontology and coreference). OntoNotes Release 5.0 contains the content of earlier releases and adds source data from and/or additional annotations for newswire, broadcast news, broadcast conversation, telephone conversation and web data in English and Chinese and newswire data in Arabic. Here we use the English dataset of OntoNotes 5.0. Table 4 shows the statistics of the datasets. We compared our model with the vanilla BERT-tagger. We trained our model for 10 and 50 epochs on the OntoNotes 5.0 and CoNLL 2003 datasets, respectively. Other hyper-parameter settings remained the same as the Chinese dataset experiments. Table 5 shows our results on the two English datasets. From the F1 results, we can see that our model is slightly worse than the baseline.

Table 4. Details of two English datasets. #ENT: number of entities; S: number of sentences; T: number of tokens.

Dataset	#ENT	Type	Train	Dev	Test
CoNLL 2003	4	S	15.0k	3.3k	3.5k
		T	204.6k	51.4k	46.4k
OnotNote 5.0	18	S	59.9k	8.5k	8.3k
		T	1088.5k	147.7k	152.7k

Table 5. Results for VPN on the two English datasets.

Method	Datasets					
	CoNLL 2003			OntoNotes 5.0		
	P	R	F1	P	R	F1
BERT-tagger	-	-	92.8	90.01	88.35	89.16
VPN (our method)	92.21	91.55	91.88	88.52	88.62	88.57

The label-word selection procedure is of great importance in our model. We collect the label words and their weights in the raw datasets, with the label words being natural-language words. Note that the natural-language words cannot be predicted in pre-trained models, so we need to convert the natural-language label words to label tokens in the vocabulary of the pre-trained model. In the Chinese language, the smallest unit of text is a character, and the tokens in the vocabulary of a pre-trained model are almost all characters. For example, when we feed the input sentence {“今天出发去上海”} (Today, I’m leaving for Shanghai) the tokenized output using the `bert-base-chinese` (https://github.com/google-research/bert, accessed on 20 March 2023) pre-trained model is {“今”, “天”, “出”, “发”, “去”, “上”, “海”} (in the Chinese language, the phrase “今天” means "today”; “出发” means “leave”; “去上海” means “go to Shanghai”) . We can see that the natural-language input sentence and output tokens are almost the same, with the output tokens still retaining the semantics of the input sentence. However, things are different when it comes to the English language. For the English language, the tokenizers of the pre-trained model tend to split the natural word into its sub-words. For example, the word *miscellaneous* expresses clear semantics, while the tokenized result *mi*, *##s*, *##cell*, *aneous* loses the original semantics of *miscellaneous*. Therefore, the reason we cannot obtain the best performance is probably because the tokenization procedure is more complex for the English language, so even if we find suitable natural-language label words, it is still difficult for us to find suitable words in the vocabulary of the pre-trained model to express the semantics hidden in the entity labels.

For future work, we will explore better label-word selection methods to find suitable tokens in the vocabularies of the pre-trained models to better express the semantics of tags. In the English dataset, we can choose not to use words that can be split into sub-words by tokenizers as our label words. Furthermore, we will explore generative pre-trained models, such as GPT-3, as our backbone model and let the model predict the label words.

6.3. Results per Entity Type

In Figures 6–10, we draw the confusion matrix head maps using sklearn [33]. Meanwhile, we report the results per entity class. Tables 6–10 are the experimental results of Weibo NER, MSRA, OntoNotes 4.0, CoNLL 2003, and OntoNotes 5.0, respectively. From the results, we find that the scores of big datasets such as MSRA and OntoNotes 4.0 are much better than those of small datasets such as Weibo NER. Moreover, we can see that the total entity num of a specific entity type affects the results a lot: in line 3 and line 5 of Table 6, the scores of entity type `LOC` are much less than those of `PER`, and in line 3 and line 5 of Table 8, the scores of entity type `LOC` are also much less than those of `PER`. Furthermore, we notice that `ORG` entities are more likely to be predicted as the `GPE` entity type compared to other entities in Figure 6, and vice versa. This may be because the semantic information of those two entity types is very close, and it can be hard to find suitable label words to distinguish them. In the OntoNotes 4.0 dataset, we find that the entity num of `ORG` and `PER` are very close in Table 8, but the scores of `ORG` are much less than those of `PER`. Observing Figure 8, we can see that there are occurrences of misidentification between the `GPE`, `LOC`, and `ORG` entity types, which may indicate that designing these three difficult-to-distinguish entity types in the same dataset is unwise. In the CoNLL 2003 dataset, the entity num of `LOC` is much fewer than that of other entity types, and accordingly, its performance scores were notably lower than those of the other entity types. This suggests that a larger

number of entities is required to provide adequate training for the model, resulting in an improved performance. In the big OntoNotes 5.0 dataset, which has 18 entity classes, the entity distribution is unbalanced. In Table 10, we can see that the scores of entity types with a small portion of the total entity num are much less than those of entity types with a big portion of the total entity num.

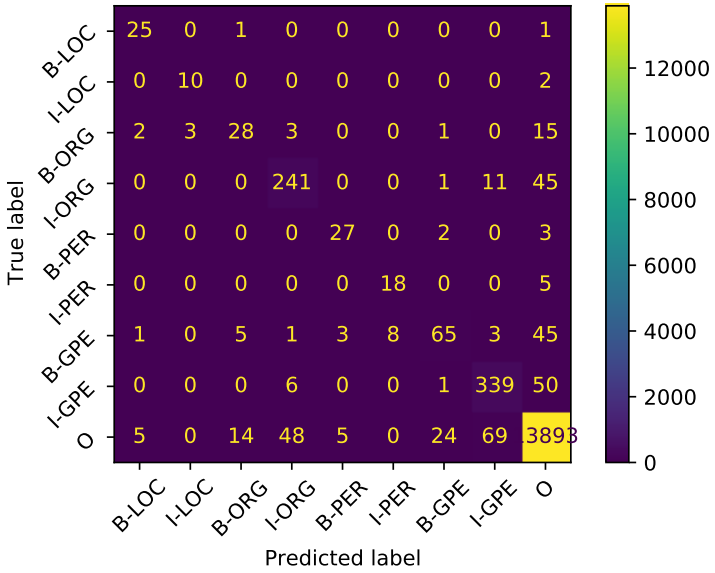

Figure 6. Confusion matrix on Weibo NER dataset. The number on the bottom right is 13,893.

Table 6. Results of Weibo NER per entity class.

Entity Type	Entity Num	P	R	F1
GPE	60	70.00	85.71	77.06
LOC	30	50.00	53.57	51.72
ORG	44	63.64	50.00	56.00
PER	289	77.16	78.25	77.70
Total	423	78.81	73.68	73.25

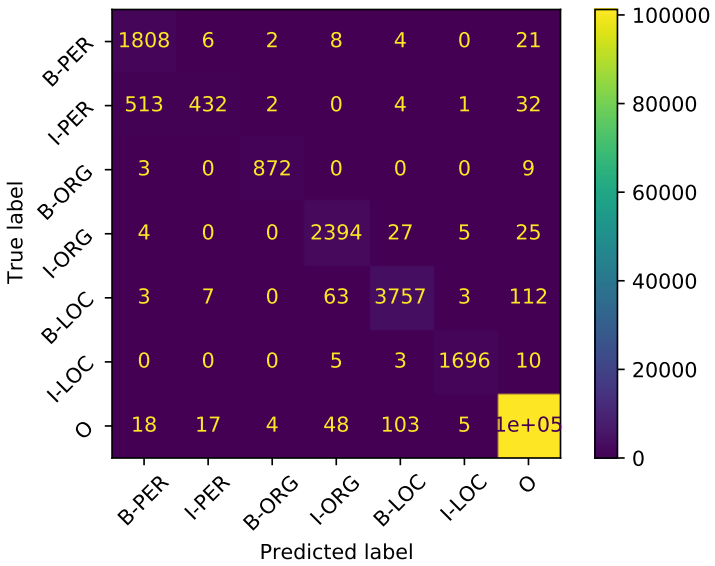

Figure 7. Confusion matrix on MSRA dataset. The number on the bottom right is 101,228.

Table 7. Results of MSRA per entity class.

Entity Type	Entity Num	P	R	F1
LOC	3471	97.44	95.78	96.60
ORG	2203	93.37	94.14	93.76
PER	1859	98.39	98.12	98.25
Total	7523	96.48	95.88	96.18

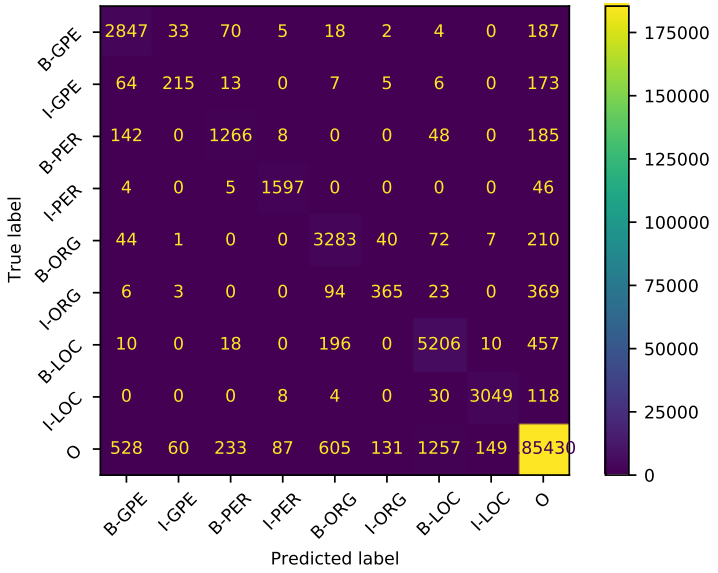

Figure 8. Confusion matrix on OntoNotes 4.0 dataset. The number on the bottom right is 185,430.

Table 8. Results of OntoNotes 4.0 per entity class.

Entity Type	Entity Num	P	R	F1
GPE	3765	80.66	87.98	84.16
LOC	421	53.92	46.23	49.78
ORG	1954	73.75	76.77	75.25
PER	1962	91.95	96.78	94.30
Total	8102	80.34	84.71	82.47

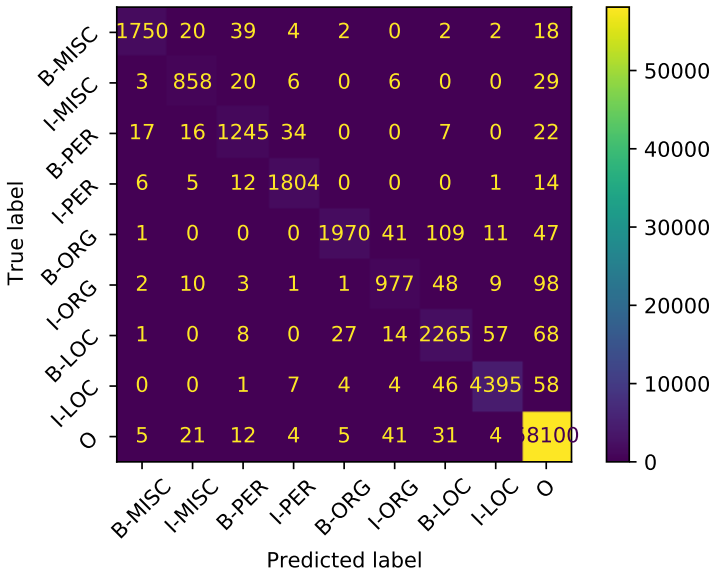

Figure 9. Confusion matrix on CoNLL 2003 dataset. The number on the bottom right is 58,100.

Table 9. Results of CoNLL 2003 per entity class.

Entity Type	Entity Num	P	R	F1
LOC	1606	95.21	91.67	93.40
MISC	711	82.84	83.90	83.37
ORG	1660	90.30	90.25	90.27
PER	1631	95.28	96.10	95.69
Total	5608	91.21	91.55	91.88

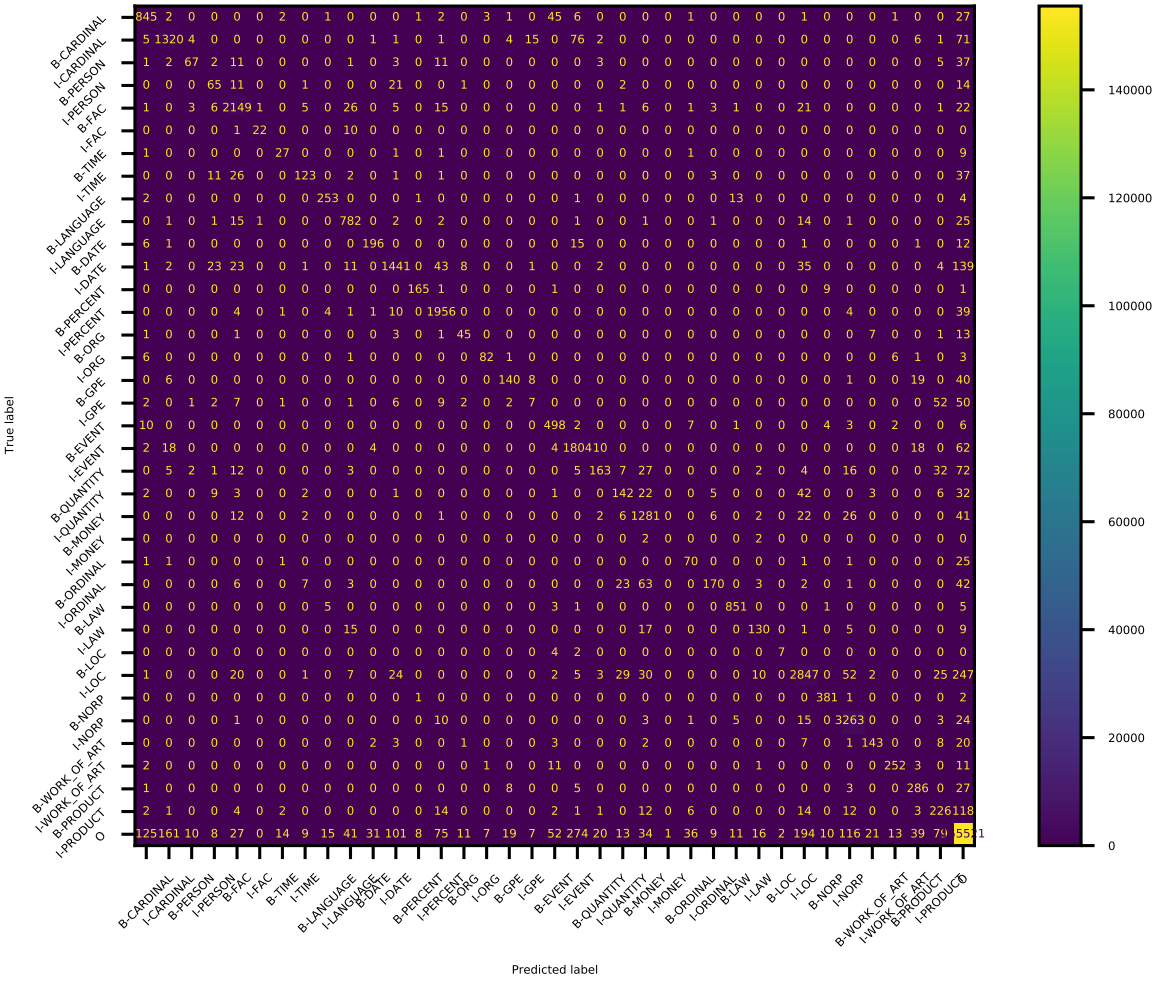

Figure 10. Confusion matrix on OntoNotes 5.0 dataset. The number on the bottom right is 155,521.

Table 10. Results of OntoNotes 5.0 per entity class.

Entity Type	Entity Num	P	R	F1
CARDINAL	946	84.46	85.45	84.95
DATE	1,663	84.73	87.95	86.31
EVENT	59	71.19	66.67	68.85
FAC	148	69.59	76.30	72.79
GPE	2,191	95.85	93.75	94.79
LANGUAGE	19	78.95	68.18	73.17
LAW	51	63.41	65.00	64.20
LOC	180	68.89	69.27	69.08
MONEY	318	91.19	92.36	91.77
NORP	857	92.42	94.17	93.29

Table 10. *Cont.*

Entity Type	Entity Num	P	R	F1
ORDINAL	216	80.56	89.23	84.67
ORG	1,742	87.14	84.57	85.84
PERCENT	353	92.35	93.41	92.88
PERSON	2,005	94.01	94.82	94.42
PRODUCT	71	73.24	68.42	70.75
QUANTITY	110	79.09	82.86	80.93
TIME	215	65.58	66.51	66.04
WORK_OF_ART	136	68.38	56.02	61.59
Total	11,270	88.52	88.62	88.57

7. Conclusions

In this work, we proposed a simple yet effective variation on prompt tuning for Chinese NER. We took the one-pass decoding strategy, which significantly increases the decoding speed. We let the LM predict several label words derived from a training dataset and convert them to label tokens in the vocabulary of the pre-trained model, retrieving the overall tag-related logits. These label words are more relevant to the tag than the classes; in this way, we can also model the logits of positions labelled `O` and use the `BIO` scheme rather than the `IO` scheme, which can use the CRF layer to boost the model's performance. Experiments show that our proposed method outperforms state-of-the-art models for three popular datasets. For small datasets, such as Weibo, compared to the vanilla BERT-tagger, the rest of the baseline models have little improvement, while our model improved by 4–5%.

Author Contributions: Conceptualization, N.H., X.Z., B.X., X.X., H.L. and H.-T.Z.; methodology, N.H., X.Z. and B.X.; software, B.X., N.H. and X.Z.; investigation, N.H., X.Z., B.X. and X.X.; data curation, X.Z.; writing—original draft preparation, N.H.; writing—review and editing, N.H., X.Z. and X.X.; visualization, N.H.; X.Z., B.X., H.L., X.X. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China(Grant No.62276154), Research Center for Computer Network(Shenzhen) Ministry of Education, Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No.2023A1515012914), Basic Research Fund of Shenzhen City (Grant No.JCYJ20210324120012033 and JSGG20210802154402007), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (HW2021008).

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and data are released at https://github.com/huniu20/vpn; we cannot provide the OntoNotes 4.0 dataset due to privacy issues, but it can be found at https://www.ldc.upenn.edu/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
2. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
3. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
4. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **2022**, *3*, 182–192. [CrossRef]

5. Schick, T.; Schütze, H. 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, Online, 19–23 April 2021; pp. 255–269.
6. Tam, D.; Menon, R.R.; Bansal, M.; Srivastava, S.; Raffel, C. Improving and Simplifying Pattern Exploiting Training. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 September 2021; pp. 4980–4991.
7. Perez, E.; Kiela, D.; Cho, K. True few-shot learning with language models. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11054–11070.
8. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 September 2021; pp. 3045–3059.
9. Qin, G.; Eisner, J. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Online, 6–11 June 2021.
10. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT Understands, Too. *arXiv* **2021**, arXiv:2103.10385.
11. Zhong, Z.; Friedman, D.; Chen, D. Factual Probing Is [MASK]: Learning vs. Learning to Recall. *arXiv* **2021**, arXiv:2104.05240.
12. Cui, L.; Wu, Y.; Liu, J.; Yang, S.; Zhang, Y. Template-Based Named Entity Recognition Using BART. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 1835–1845.
13. Ma, R.; Zhou, X.; Gui, T.; Tan, Y.; Li, L.; Zhang, Q.; Huang, X.J. Template-free Prompt Tuning for Few-shot NER. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online and Seattle, WA, USA, 10–15 July 2022; pp. 5721–5732.
14. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
15. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? 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), Hong Kong, China, 3–7 November 2019; pp. 2463–2473.
16. Davison, J.; Feldman, J.; Rush, A.M. Commonsense Knowledge Mining from Pretrained 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), Hong Kong, China, 3–7 November 2019; pp. 1173–1178.
17. Trinh, T.H.; Le, Q.V. A Simple Method for Commonsense Reasoning. *arXiv* **2018**, arXiv:1806.02847.
18. Schick, T.; Schütze, H. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8766–8774. [CrossRef]
19. Schick, T.; Schmid, H.; Schütze, H. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 5569–5578.
20. Gao, T.; Fisch, A.; Chen, D. Making Pre-trained Language Models Better Few-shot Learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 3816–3830.
21. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 4582–4597.
22. Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; et al. *Ontonotes Release 4.0*; LDC2011T03; Linguistic Data Consortium: Philadelphia, PA, USA, 2011.
23. Che, W.; Wang, M.; Manning, C.D.; Liu, T. Named entity recognition with bilingual constraints. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, 9–15 June 2013; pp. 52–62.
24. Zhang, S.; Qin, Y.; Hou, W.J.; Wang, X. Word segmentation and named entity recognition for sighan bakeoff3. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Sydney, Australia, 22–23 July 2006; pp. 158–161.
25. Peng, N.; Dredze, M. Named entity recognition for chinese social media with jointly trained embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 548–554.
26. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 282–289.
27. Meng, Y.; Wu, W.; Wang, F.; Li, X.; Nie, P.; Yin, F.; Li, M.; Han, Q.; Sun, X.; Li, J. Glyce: Glyph-vectors for chinese character representations. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 2746–2757. .
28. Li, X.; Yan, H.; Qiu, X.; Huang, X.J. FLAT: Chinese NER Using Flat-Lattice Transformer. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6836–6842.
29. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5849–5859.

30. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015, pp. 649–657 .
31. Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv* **2003**, arXiv:cs/0306050.
32. Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H.T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; Zhong, Z. Towards robust linguistic analysis using ontonotes. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, Sofia, Bulgaria, 4–9 August 2013; pp. 143–152.
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.