applied sciences

MDPI

*Article*

# A Chinese Few-Shot Text Classification Method Utilizing Improved Prompt Learning and Unlabeled Data

**Tingkai Hu [1], Zuqin Chen [2,*], Jike Ge [1], Zhaoxu Yang [1] and Jichao Xu [1]**

[1] College of Intelligent Technology and Engineering, Chongqing University of Science and Technology, No. 20 University East Road, Chongqing 401331, China
[2] School of Library, Chongqing University of Science and Technology, No. 20 University East Road, Chongqing 401331, China
* Correspondence: chenzuq81@163.com

**Abstract:** Insufficiently labeled samples and low-generalization performance have become significant natural language processing problems, drawing significant concern for few-shot text classification (FSTC). Advances in prompt learning have significantly improved the performance of FSTC. However, prompt learning methods typically require the pre-trained language model and tokens of the vocabulary list for model training, while different language models have different token coding structures, making it impractical to build effective Chinese prompt learning methods from previous approaches related to English. In addition, a majority of current prompt learning methods do not make use of existing unlabeled data, thus often leading to unsatisfactory performance in real-world applications. To address the above limitations, we propose a novel Chinese FSTC method called CIPLUD that combines an improved prompt learning method and existing unlabeled data, which are used for the classification of a small amount of Chinese text data. We used the Chinese pre-trained language model to build two modules: the Multiple Masks Optimization-based Prompt Learning (MMOPL) module and the One-Class Support Vector Machine-based Unlabeled Data Leveraging (OCSVM-UDL) module. The former generates prompt prefixes with multiple masks and constructs suitable prompt templates for Chinese labels. It optimizes the random token combination problem during label prediction with joint probability and length constraints. The latter, by establishing an OCSVM model in the trained text vector space, selects reasonable pseudo-label data for each category from a large amount of unlabeled data. After selecting the pseudo-label data, we mixed them with the previous few-shot annotated data to obtain brand new training data and then repeated the steps of the two modules as an iterative semi-supervised optimization process. The experimental results on the four Chinese FSTC benchmark datasets demonstrate that our proposed solution outperformed other prompt learning methods with an average accuracy improvement of 2.3%.

**Keywords:** Chinese few-shot text classification; prompt learning; unlabeled data; pre-trained language model

check for updates

**Citation:** Hu, T.; Chen, Z.; Ge, J.; Yang, Z.; Xu, J. A Chinese Few-Shot Text Classification Method Utilizing Improved Prompt Learning and Unlabeled Data. *Appl. Sci.* **2023**, *13*, 3334. https://doi.org/10.3390/app13053334

Academic Editor: Kuei-Hu Chang

Received: 5 February 2023
Revised: 25 February 2023
Accepted: 2 March 2023
Published: 6 March 2023

## 1. Introduction

Text classification is a vital task of natural language processing (NLP) that involves assigning text data to predefined categories or labels, such as emotions, topics, and other types [1]. It is widely used in practical applications such as sentiment analysis [2–4], information retrieval [5–7], and question-answering [8]. However, the traditional text classification models require large, well-labeled datasets, which can be challenging to obtain in practice. Insufficiently labeled samples and low-generalization performance becomes in-creasingly serious as the text label types grow. To tackle this challenge, researchers have turned to the few-shot text classification (FSTC), which aims to solve the problem by limiting the number of labeled data.

To extract useful semantic information from few-shot data, the pre-trained language model (PLM) is usually adopted. Recently, a technique known as prompt learning [9–11] has shown impressively in FSTC tasks. This method utilizes natural language prompts and the PLM and converts downstream tasks into masked language modeling problems. Compared to traditional approaches, prompt learning has proven to be particularly effective in extreme scenarios where there is only one training sample per type. Although much research has been conducted on prompt learning for English FSTC, its application in Chinese FSTC is still in the early stages, given the language disparity between Chinese and English. Prompt learning methods require a pre-trained language model and token of the vocabulary list for training, and various language models have varying token coding structures, making it difficult to adopt previous English-related approaches for effective Chinese prompt learning. For example, the label "computer" requires three tokens in Chinese, while it only needs a single token in English. This is because Chinese is a highly context-dependent language. Meaningful labels often require multiple tokens to express the full meaning and context of the label, whereas, in English, a single token can often convey the same meaning. Thus, how to design a generic prompt template for Chinese labels has become the first problem to be solved. In addition, the Chinese language characteristic results in the mask tokens being expanded to more than one when prompt learning is used. This may be a random token composition problem with the prediction labels obtained by mask decoding. Random token composition does not belong to a predefined label set, which produces an output result that is difficult to accept. Hence, how to deal with a random token composition when predicting Chinese labels is the second problem to be solved. Additionally, with the unprecedented volume of unlabeled data generated from the practice environment, recent research on how to effectively utilize existing, unlabeled datasets within prompt learning has attracted significant attention. On the one hand, unlabeled data are easier to obtain than labeled data; on the other hand, overfitting from classification previously can be avoided to a significant extent. Semi-supervised learning using unlabeled data has emerged as an important approach to classification, data mining, and information retrieval. Pseudo-label paradigms that assign labels with high confidence to unlabeled data based on a trained model have been increasingly investigated [12]. With more unlabeled data, better classification performance can be expected by increasing the size of the trainable data. However, the convergence of prompt learning and unlabeled data is a challenging research topic that has not been extensively studied.

For the issues discussed above, we propose a Chinese few-shot text classification method called CIPLUD that combines an improved prompt learning method and existing unlabeled data for Chinese FSTC. The CIPLUD model is composed of two modules, including the Multiple Masks Optimization-based Prompt Learning (MMOPL) module and the One-Class Support Vector Machine-based Unlabeled Data Leveraging (OCSVM-UDL) module. As part of the MMOPL module, our method designs universal prompt templates with multiple masks for different tasks. We then built a text classification model using universal prompt templates and a Chinese pre-trained language model. Afterward, we optimized the predicted label of the model using joint probability and length constraints, effectively eliminating the problem of a random token composition. In the OCSVM-UDL module, our method assigns pseudo-labels to the unlabeled data through a one-class support vector machine model [13] inspired by anomaly detection. With the help of few-shot-labeled data, the OCSVM model can obtain spherical constraint boundaries for different classes in the feature space through training. Then, these constraint boundaries are used to filter the unlabeled data. The unlabeled samples that do not fall into any boundary are considered low-confidence anomaly data, and the unlabeled samples that fall into multiple constraint boundaries are considered ambiguous data. This filtering approach can filter noise data from the unlabeled data and effectively screen out pseudo-labeled data with high confidence and disambiguation. Finally, the new training data are created by blending the pseudo-labeled data with the few-shot-labeled data. The flow of the above-mentioned two modules is repeated until the performance of the text classification model stabilizes, signaling the end of the iteration. The experimental results on the few-shot text classification datasets, FewCLUE [14], demonstrate the effectiveness of our approach. The proposed model is referred to as the CIPLUD model. The main contributions of our paper are summarized as follows:

- According to our best knowledge, this is the first study to combine the prompt learning method and unlabeled data for Chinese FSTC. The proposed CIPLUD model de-signs a universal prompt template for different Chinese FSTC tasks. It uses the joint probability and length-constrained decoding method to solve the random token composition problem caused by the multiple masks' prompt templates;
- To design a pseudo-label candidate method using the OCSVM model and conduct semi-supervised training using the pseudo-labels, resulting in improved performance;
- We conducted experiments and evaluated a series of Chinese FSTC datasets. The experimental results demonstrate that the proposed CIPLUD model can gain significant improvement over other prompt learning methods.

The rest of the paper is organized as follows. We review some representative work on the few-shot text classification and prompt learning in Section 2. In Section 3, we describe the architecture of the proposed model in detail. Sections 4 and 5 introduce the experiments on the FewCLUE dataset and present the experimental results and analysis to demonstrate the competitiveness of the proposed method compared to the most advanced prompt learning models for the few-shot text classification. Finally, we provide conclusions and discuss some future research directions in Section 6.

## 2. Related Work

In this section, we review some representative work on the few-shot text classification, pre-trained language models, and prompt learning methods.

### *2.1. Few-Shot Text Classification*

Few-shot text classification aims to build a classifier using a limited number of annotated resources. Traditional text classification methods, such as the Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Transformers, are not well suited for this scenario as they require a large number of labeled samples for model convergence [15–19]. For the past few years, the meta-learning approach has emerged as a popular framework for few-shot text classification, which aims to help us solve tasks by learning a set of universal "meta-tasks" with few samples. There are two main ways to leverage meta-learning for target tasks: (1) the optimization-based method and (2) the metric-based method. The former treats the meta-task as a universal parameter optimization process and its modification in a large number of similar tasks. The latter focuses on measuring the similarity between the query and support samples by generating sample representations with better clustering properties, which are then used to model the classification probability of the query samples [20].

Several studies have demonstrated the effectiveness of these meta-learning ap-proaches in few-shot text classification [21]. Metric-based meta-learning, especially, can outperform traditional text classification models trained from scratch. For example, Koch et al. [22] proposed Siamese Neural Networks, a neural network model containing two identical neural networks, to solve single-sample learning problems by discriminating whether the query and support samples belong to the same class. Inspired by the twin networks, Vinyals et al. [23] proposed matching networks, which use global information within each episode to represent the query samples and then use an attention mechanism to weigh the aggregated label information of the support samples to model the classification probability of the query samples. Snell et al. [24] proposed a simple and effective few-shot learning algorithm called the Prototype Network to form a circular representation of this category and then measure the class works, averaging the sample representations of the same category in the support samples' classification probability of the query samples by their similarity to each prototype representation. Furthermore, Sung et al. [25] proposed a Relation Network to model the similarity metric function using deep neural networks to replace the traditional fixed distance calculation methods, such as the L1 distance, L2 distance, and cosine similarity.

Although metric-based meta-learning creates insufficiently labeled sample scenarios for each meta-task through clustering characteristics and text similarity, they still need to filter and retain a large amount of representative annotation data for training and debugging.

### 2.2. Pre-Trained Language Model

Pre-trained language models play a pivotal role in text classification, serving as the backbone for models in both sufficient data and few-shot scenarios. There are two prominent examples of this type of model: masked language models (MLM) and left-to-right (LTR) language models [26]. Masked language models use a technique called "masking", which means that a certain percentage of words in a given sentence is randomly replaced with a placeholder symbol, such as BERT [27] and ERNIE [28]. This type of model attempts to use the surrounding context to predict what the missing word could be. Left-to-right language models are designed to assign a probability to a sequence of words and can be applied to generate more natural language, such as GPT [29]. By leveraging these powerful pre-trained models, text classification models can be rapidly adapted to new tasks with only a minimal amount of annotated data.

### 2.3. Prompt Learning

Prompt learning is a popular approach in text classification that leverages the power of pre-trained language models [30]. In this approach, a model is fine-tuned using manually crafted or automatically generated prompt sentences with mask tokens added to the original text. The goal is to use the pre-trained model to predict the tokens at the mask lo-cations, which are associated with the labels in the text classification task. According to different prompt designs, there are two main types of prompt learning: discrete and con-tinuous prompts. In discrete prompt learning, a set of natural language prompt tokens is used to classify the text, such as PET [31]. For example, given an input text, the input em-bedding sequence can be formulated with the input text and the prompt tokens, such as "It is a [MASK] subject". However, the discrete prompt is a locally optimal process because the neural network is continuous, and the search is conducted in a discrete space. In contrast, continuous prompt learning considers that the prompt templates can be trainable and optimized for continuous prompt embeddings. This method trains the input sequence with the input text, a series of trainable, continuous embeddings, and is a placeholder for the model to predict the mask token. For instance, the EFL [32] approach uses the T5 model to generate optimal discrete prompt templates, eliminating the need for a manual search. Another method, P-tuning [33,34], considers that the prompt templates can be trainable and optimized for continuous prompt embeddings, achieving comparable performance to the BERT fine-tuning in a supervised learning process.

Prompt learning methods often make use of a mask token in their templates, linking each label to a verbalization token to predict label categories. Despite its effectiveness, it is rarely suitable for tasks with complex label spaces and labels of varying lengths, especially Chinese labels that need multiple tokens to convey meaning.

## 3. Chinese Few-Shot Text Classification Method with Improved Prompt Learning and Unlabeled Data

In this section, we describe the proposed CIPLUD model, which can be broken down into two components: the Multiple Masks Optimization-based Prompt Learning (MMOPL) module and the One-Class Support Vector Machine-based Unlabeled Data Leveraging (OCSVM-UDL) module. In Section 3.1, we discuss the MMOPL module, highlighting how it formulates a general prompt learning template and optimizes predictive labels for superior performance in text classification tasks. Section 3.2 delves into the workings of the OCSVM-UDL module, demonstrating how these two modules collaborate to achieve improved outcomes in text classification.

### *3.1. Multiple Masks Optimization-Based Prompt Learning Module*

Prompt learning is an exciting development in the field of natural language processing, offering great promise in few-shot text classification tasks. At our lab, we have endeavored to enhance the effectiveness of prompt learning in the context of Chinese text classification by introducing the Multiple Masks Optimization-based Prompt Learning (MMOPL) module, as depicted in Figure 1.

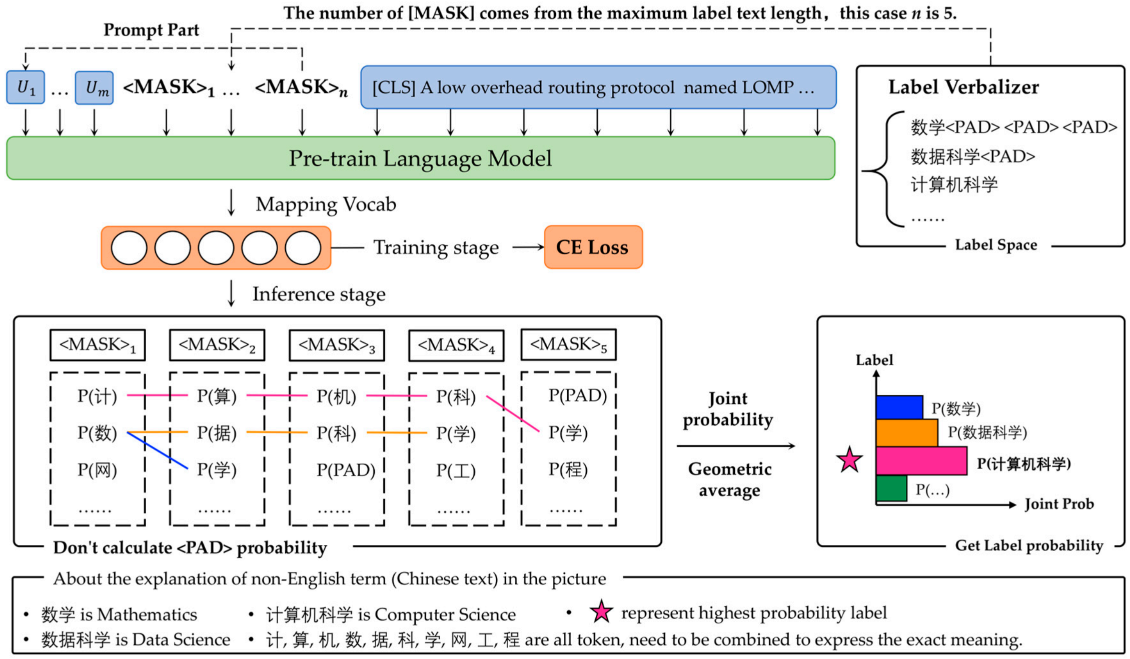
The number of [MASK] comes from the maximum label text length，this case $n$ is 5.
Prompt Part
$U_1$ … $U_m$ <MASK>$_1$ … <MASK>$_n$
[CLS] A low overhead routing protocol named LOMP …
Pre-train Language Model
Mapping Vocab
Training stage
CE Loss
Inference stage
Label Verbalizer
数学<PAD> <PAD> <PAD>
数据科学<PAD>
计算机科学
……
Label Space
<MASK>$_1$
<MASK>$_2$
<MASK>$_3$
<MASK>$_4$
<MASK>$_5$
P(计) P(数) P(网) ……
P(算) P(据) P(学) ……
P(机) P(科) P(PAD) ……
P(科) P(学) P(工) ……
P(PAD) P(学) P(程) ……
Don't calculate <PAD> probability
Joint probability
Geometric average
Label
P(数学)
P(数据科学)
P(计算机科学)
P(…)
Joint Prob
Get Label probability
About the explanation of non-English term (Chinese text) in the picture
• 数学 is Mathematics
• 数据科学 is Data Science
• 计算机科学 is Computer Science
• 计, 算, 机, 数, 据, 科, 学, 网, 工, 程 are all token, need to be combined to express the exact meaning.
• ★ represent highest probability label

**Figure 1.** The few-shot text classification model is constructed through the Multiple Masks Optimization-based Prompt Learning module. The rose star symbol (★) in the image represents the label with the highest probability of scoring among all the labels predicted by the model.

In prompt learning, there are two primary engineering steps, namely prompt engineering and answer engineering. Prompt engineering involves designing a prompt function, $F_{prompt}(X)$, that elicits the most optimal performance in downstream tasks. In the past, discrete prompts that are created through a painstaking manual process have been used. However, this approach is both time-consuming and requires considerable expertise, and even experienced prompt designers may not be able to create the best prompt [35]. To overcome this limitation, we propose the use of continuous prompts that are automatically learned by the model. These prompts are tensors designed to enable the effective execution of the tasks by the language model, and they do not necessarily have to be limited to human-understandable natural language. Continuous prompts are advantageous because they eliminate the need for extensive time and effort spent on manual searches and the adjustment of discrete prompts [36].

To further enhance the effectiveness of prompt learning for Chinese text classification tasks, we have introduced the use of multiple masks in the MMOPL module. This approach involves creating a set of learnable tensors, $[U_1, \ldots, U_m]$, where “m” refers to the number of learnable tensors for each original text, X, as continuous prompts. We then concatenate the mask token, which corresponds to the label that requires classification. By formalizing this into a prompt template, $[U_1, \ldots, U_m, E(\text{“[MASK]”})]$, we enable the creation of a prompt that is effective for the specific task at hand.

To achieve optimal results in prompt learning, we have improved the mask token and answer engineering. Specifically, we have developed an adaptive method for expanding the mask token to the required number when generating the input sequence. To achieve this, we first calculate the maximum length value, *n*, of the label text in the label set. This value is then used as the number of MASK tokens. For labels with lengths that are less than n, we use the [PAD] placeholder to fill in the gaps. By concatenating the continuous prompts with multiple masks and the original sentence, E(x), we create an embedded input sequence, = [$U_1$, . . . , $U_i$, E("[MASK]$_1$") . . . E("[MASK]$_n$"), E(X)]. Here, E("[MASK]$_1$") . . . E("[MASK]$_n$") are mask tokens that are adaptively expanded according to the label, and E(X) is the original text. An important aspect to consider about the prompt template is its placement concerning the original text, which is contingent on the choice of a pre-trained model. In our case, we have chosen to use ERNIE, the pre-trained model that has a maximum input length of 512 characters. To ensure that the prompt is not truncated, we have opted to employ the prefix concatenation method.

During the training stage, assuming the output of the Chinese pre-trained language model is $O \in \mathrm{R}^{|X|*d}$, the text data are fed into the Chinese pre-trained model.

$$v = F_{\text{prompt}}(\mathrm{X}) = [\mathrm{U_1},\ \ldots,\ \mathrm{U_i},\ \mathrm{E}(\text{"[MASK]}_1\text{"})\ \ldots\ \mathrm{E}(\text{"[MASK]}_n\text{"}),\ \mathrm{E}(\mathrm{X})] \tag{1}$$

$$O_{[mi]} = PLM(v) \tag{2}$$

where $v$ is the input text representation. The label probability distribution of the [MASK] tokens can be obtained by using a full-connection layer with the activation function:

$$P_m = softmax\left(O_{[m]} * W_m + b_m\right) \tag{3}$$

where $b_m$ is a bias of the weight of the full-connection layer, and $W_m \in \mathbb{R}^{d*|V|}$ is the weight of the full-connection layer. d is the hidden layer size of the pre-trained language model, and |V| is the vocab size of the pre-trained language model. $O_{[m]}$ refers to the selection of a tensor based on the indices of the label tokens. Then, we use the Cross-Entropy (CE) as the target loss function and exclude the loss from the [PAD] in the calculation of the loss function, which can be written as follows:

$$\mathcal{L} = \begin{cases} \frac{1}{|N|}\sum_{i=1}^{|N|}[t_i = j]\log\left(P_{m(i,j)}\right), & |j \neq [PAD] \\ 0, & |j = [PAD] \end{cases} \tag{4}$$

where $t_i$ is the ground truth label of Sample $i$, and $j$ denotes the category of the label. $N$ is the number of samples, and $P_{m(i,j)}$ is the probability that the $i$th sample in the prediction probability distribution belongs to category $j$.

During the inference stage, effective answer engineering is crucial for developing an accurate prediction model. Unlike prompt engineering, which is focused on ensuring the appropriate input for prompt learning, answer engineering is responsible for mapping the answer space, Z, to the original output, Y. In many cases, the answer space includes all tokens in the pre-trained vocabulary. Typically, a mask token is mapped to a label token with the highest probability score in the pre-trained vocabulary. This method works well in scenarios where there is only a single mask token in the prompt learning. However, when the mask token is expanded to multiple tokens, mapping each mask token to the token with the highest probability score can lead to random token combinations. This is because there is no semantic connection or constraint between the output tokens during this process, which makes it difficult to accept in actual prediction scenarios. When the mask token is expanded to multiple tokens, it may lead to random token combinations if we use the traditional approach of mapping each mask token to the token with the highest probability score. To address this problem, we use joint probability to calculate the overall probability score of each label token corresponding to the mask position. We then select the label with the highest overall probability score as the final result, Y, thereby completely avoiding the issue of random token combinations.

In addition to solving the issue of random token combinations, we must also tackle the challenge of longer labels struggling to compete in the probability due to the problem of probability vanishing. In Chinese text, labels do not have a fixed length, and longer text often leads to smaller joint probabilities, making it difficult for lengthy labels to emerge as winners in the probability competition. To address this issue, we introduce the use of the geometric mean to constrain the joint probability calculation of the labels. The root of the joint probability is then taken to the M-power, resulting in the constrained joint probability, as shown in the function Formula (5), which includes length constraints.

$$P_M = \sqrt[M]{\prod_{i=1}^{M} P\big(E[MASK]_i \mid E[MASK]_1 \cdots E[MASK]_{i-1}\big)} \tag{5}$$

Lastly, we perform length constraint processing on all the joint probability labels and use an activation function to normalize the probability. The predicted label with the highest joint probability value is selected as the final result, Y, ensuring that the answer engineering process is both efficient and accurate.

### *3.2. One-Class Support Vector Machine-based Unlabeled Data Leveraging Module*

Numerous studies have demonstrated that utilizing unlabeled data can effectively enhance the few-shot text classification [37]. Previous methods utilized trained models to provide pseudo-labels to unlabeled data, and high thresholds were used to avoid noisy data. However, this approach was time-consuming, and there was a risk of model overconfidence. Inspired by anomaly detection [38,39], we consider the potential overconfidence in unlabeled data as an anomalous phenomenon. To solve this problem, we use the One-Class Support Vector Machine (OCSVM) algorithm to create multiple spherical constraint boundaries for existing samples with different labels. These constraint boundaries are used to filter out anomalous unlabeled data and assign appropriate pseudo-labels. The spherical shape boundary is superior for text classification [40]. As shown in Figure 2, we have developed an OCSVM-based unlabeled data utilization module (OCSVM-DL).

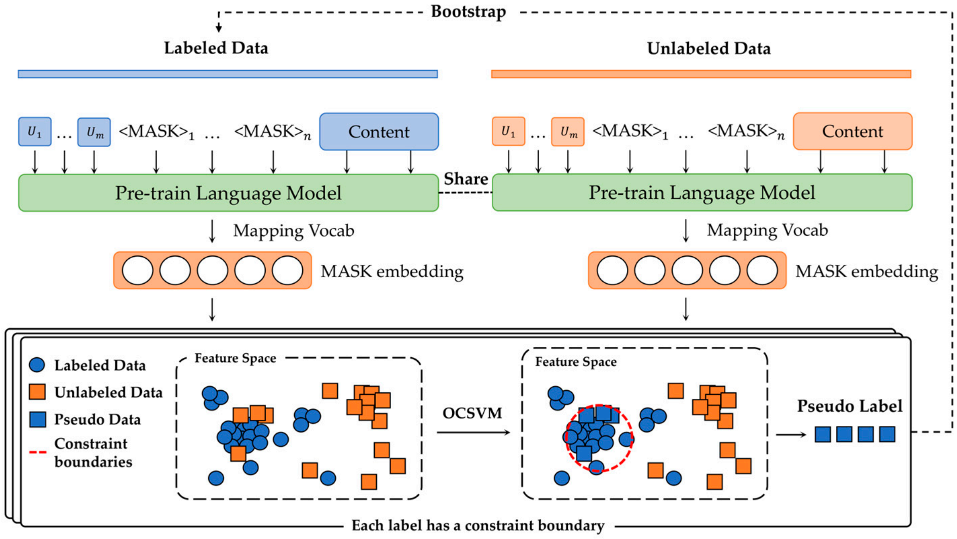

**Figure 2.** The strategy of adding unlabeled data and performing iterative training is implemented through the One-Class Support Vector Machine-based Unlabeled Data Leveraging module.

The OCSVM algorithm is designed specifically for one-class classification, and it only needs to be trained on data from a single class. Unlike the standard SVM algorithm, which aims to construct a generalized optimal classification plane that maximizes the margin between the two classes, the OCSVM seeks to find a hyper-sphere in the feature space that contains all the data and has the smallest possible volume, maximizing the margin between the inlier data and the outliers. The OCSVM optimization problem can be formulated as follows:

$$\underset{\xi \in R^{i}, \rho \in R}{min} \qquad \frac{1}{2}\|\omega^{2}\| + \frac{1}{\mu l}\sum\nolimits_{i=1}^{1} \xi_{i} - \rho \tag{6}$$

$$\text{s.t.} \quad \varPhi(O_{i})\omega \geq \rho - \xi_{i}, \xi_{i} \geq 0 \tag{7}$$

where $O_i$ is the input text feature vector, $l$ is the number of trained samples, $\varPhi$ is the feature map function that maps the input to the feature space, $\omega$ and $\rho$ are the normal vector and offset of the hyper-sphere, $\xi_i$ are the slack variables that denote the coefficient of the regular term, and $\mu \in (0,1)$ is a hyper-parameter that controls the upper bound of the error sample ratio in the total sample. If $\omega$ and $\rho$ are the solutions to the optimization problem, the decision function is:

$$\mathrm{f(x)} = \mathrm{sgn}(\ \varPhi(\mathrm{x})\omega - \rho) \tag{8}$$

where, for most of the feature vector points in the data set, the value should be positive, and $\omega$ is relatively small. Thus, by adjusting the hyper-parameter, $\mu$, we can control the compactness of the constraint boundaries, which is crucial in the process of assigning suitable pseudo-labels to the unlabeled data. To assign suitable pseudo-labels to the unlabeled data, we establish two filtering rules to identify the data with low-confidence pseudo-labels due to model overconfidence. The first rule considers unlabeled samples outside of any boundary as low-confidence anomaly data. The second rule considers unlabeled samples within multiple constraint boundaries as ambiguous data. These filtering methods can effectively eliminate noise from the unlabeled data, screening out high-confidence and unambiguous pseudo-labeled data.

After assigning the pseudo-labels to the unlabeled data, we mix the new pseudo-labeled data with the original labeled data and repeat the training of the MMOPL text classification model. We stop the iteration when the performance of the model no longer changes. The purpose of the iterative training process is to learn more monitoring signals that combine fast learning methods and unlabeled data. In Algorithm 1, we describe the iterative training process for easier understanding. First, we treat the mixed training set as the training set (Line 1). Then, we leverage improved prompt learning to obtain an optimal text classification model (Lines 3–8). Subsequently, we filter the unlabeled data to obtain the pseudo-labels (Lines 9–12). Finally, we update the mixed training set and iterate until the text classification model converges (Lines 13–14).

**Algorithm 1** The iterative training process of the OCSVM-UDL

```
Input: Training set D, validation set D′, Unlabeled set U, Mixed Training set F, MMOPL model M1, OCSVM model M2.
1: Initialize F = D // Mixed Training set equal Training set
2: repeat
3:    repeat
4:       Load a batch size of instances B belong F and add a prompt template
5:       Generate input embedding vector using the M1 for each instance in B
6:       Update parameter by minimizing $\mathcal{L}$
7:       Save the best model $M_{1'}$ according to the average performance on D′
8:    until no more batches
9:    Load a batch size of instances B belong F and add a prompt template
10:   Generate input embedding vector using the $M_{1'}$ for each instance in B
11:   Generate a constrained boundary for each label using the M2.
12:   Filtering a batch size of instances u belong U and get pseudo-label data P
13:   Update mixed training set F = D + P and duplicate removal
14: until M1 convergence
```

## 4. Experiments

In this section, we first describe the large few-shot datasets, FewCLUE. We then report the evaluation results and compare the performance of CIPLUD with several state-of-the-art prompt learning methods.

### *4.1. Experiment Datasets*

The FewCLUE datasets, a Chinese few-shot learning evaluation benchmark, was chosen for this study. It contains nine natural language understanding tasks in Chinese, including four single sentence tasks, three sentence pair tasks, and two reading comprehension tasks. The focus of this study was on the single sentence tasks, and the detailed dataset statistics are shown in Table 1. The following datasets were used:

- EPRSTMT, an e-commerce product review dataset for sentiment analysis, with the task of predicting whether the user of the reviews is positive or negative;
- CSLDCP, a Chinese scientific literature subject classification dataset, includes 67 categories of literature labels. There are 13 broad categories, ranging from the social sciences to the natural sciences, and the task is to assign a separate category to each piece of literature through Chinese abstracts;
- TNEWS, a Toutiao short text categorized news dataset, comes from the news section of Today of Headlines. There are 15 news categories, and it is the goal of this task to categorize news titles based on their headlines;
- IFLYTEK, a long text classification dataset that collects descriptive information on various application topics related to daily life, with 119 categories. The task is to pre-dict application categories from application description information.

**Table 1.** Task description and statistics of the FewCLUE datasets.

| Subtask | Task | Train | Dev | Test Public | Unlabeled Num | Num Labels |
|---|---|---|---|---|---|---|
| EPRSTMT | SentimentAnalysis | 32 | 32 | 610 | 19565 | 2 |
| CSLDCP | LongTextClassify | 536 | 536 | 1784 | 18111 | 67 |
| TNEWS | ShortTextClassify | 240 | 240 | 2010 | 20000 | 15 |
| IFLYTEK | ShortTextClassify | 928 | 690 | 1749 | 7558 | 119 |

In this study, the quality of the classification results was evaluated using accuracy from the text classification domain. This metric was also employed in the FewCLUE report. To ensure a fair comparison with the baseline set by the FewCLUE benchmark, the same data partitioning method was followed. Specifically, each subtask features a public test set and five different sets for training and validation. The model was trained on the training sets, with the best-performing model saved based on its validation set performance. Finally, the public test sets were used to evaluate the performance, with the average of the five evaluations computed as the final result.

### *4.2. Baseline Models*

We re-implemented four baseline models for the FSTC and compared their performance with the proposed CIPLUD model. These are the descriptions of the baseline models.

- Fine-tuning is a Chinese pre-trained language model that adopts;
- Pattern-Exploiting Training (PET) employs hand-crafted templates and label words to form the prompt, along with an ensemble model to annotate an unlabeled dataset, which can be considered as a text augmentation;
- EFL uses the T5 model to generate the best discrete prompt template, eliminating the need for a manual search;
- P-tuning proposes to learn continuous prompts by inserting trainable variables into the embedded input.

To ensure the correctness of the performance measures, we used Ernie-1.0 from the FewCLUE report as the pre-trained model for all the baseline models. This allowed us to compare CIPLUD with the fine-tuning paradigm and several other recently proposed competing approaches, including PET, P-tuning, and EFL. We have summarized the differences between each method and listed them in Table 2.

**Table 2.** The difference between the baseline methods.

| Method | Prompt Designing | | Prompt Style | Use Unlabeled |
|---|---|---|---|---|
| | Templates | Mask Number | | |
| Fine-tuning | — | — | — | No |
| PET | Hand-craft | Single | Discrete | YES |
| P-tuning | Auto | Single | Continuous | No |
| EFL | Hand-craft | Single | Discrete | No |
| Ours | Auto | Multiple | Continuous | YES |

### *4.3. Implementation Details*

In our empirical study, we used Ernie 1.0 to implement our proposed model, CIPLUD. The hyper-parameters and their values can be found in Table 3. The optimal values of these hyper-parameters were set according to our best practices in the empirical study. A detailed analysis of the influence of the hyper-parameters can be found in Section 5.3.

**Table 3.** Hyper-parameters and their value in our empirical study.

| Task | Hyper-Parameters | Value | Task | Hyper-Parameters | Value |
|---|---|---|---|---|---|
| EPRSTMT | Max length | 88 | TNEWS | Max length | 33 |
| | Learning rate | $3 \times 10^{-4}$ | | Learning rate | $4 \times 10^{-5}$ |
| | $\mu$ | 0.75 | | $\mu$ | 0.65 |
| CSLDCP | Max length | 278 | IFLYTEK | Max length | 215 |
| | Learning rate | $5 \times 10^{-4}$ | | Learning rate | $7 \times 10^{-4}$ |
| | $\mu$ | 0.70 | | $\mu$ | 0.70 |

The experiments were run using PyTorch 1.8.0, Python 3.8, and Windows 10 running on a desktop computer with an Intel(R) Core(TM) i5-10600KF CPU and GeForce RTX3060 with 11 GB memory.

## 5. Discussion

### *5.1. Overall Performance*

Table 4 shows a comprehensive comparison of the proposed model, CIPLUD, and the four baseline models (fine-tuning, PET, P-tuning, and EFL) in terms of accuracy on the four FSTC tasks (EPRSTMT, CSLDCP, TNEWS, and IFLYTEK) in FewCLUE. The table presents the accuracy percentage of each model on each dataset and the average accuracy. The results show that CIPLUD has the highest accuracy among all the models, with an 85.4% accuracy on EPRSTMT, 60.4% on CSLDCP, 57.2% on TNEWS, 52.8% on IFLYTEK, and 64.0% average accuracy. CIPLUD significantly outperforms all the baseline models in terms of accuracy on all four datasets, showing its effectiveness and robustness in few-shot text classification tasks.

**Table 4.** The comparison results between our proposed model, CIPLUD, and the baselines. the Human row refers to the performance of human annotators on the tasks, and the underlined values indicate the second-best results achieved among the baseline methods. We bold-marked the highest score in each column.

| Method | EPRSTMT (Acc. %) | CSLDCP (Acc. %) | TNEWS (Acc. %) | IFLYTEK (Acc. %) | Avg (Acc. %) |
|---|---|---|---|---|---|
| Human | 90.0 | 68.0 | 71.0 | 66.0 | 73.6 |
| Fine-Tuning | 66.5 | 57.0 | 51.6 | 42.1 | 54.3 |
| PET | <u>84.0</u> | <u>59.9</u> | <u>56.4</u> | 50.3 | <u>62.7</u> |
| P-tuning | 80.6 | 56.6 | 55.9 | <u>52.6</u> | 61.4 |
| EFL | 76.7 | 47.9 | 56.3 | 52.1 | 58.3 |
| MMOPL | 82.1 | 59.8 | <u>56.4</u> | 52.2 | 62.6 |
| CIPLUD | **85.4** | **60.4** | **57.2** | **52.8** | **64.0** |

Compared to the baseline models, the advantages of the CIPLUD model can be highlighted in four few-shot text classification tasks. Firstly, the majority of the prompt learning methods demonstrate an enhanced level of performance when compared to the PLM fine-tuning methods. However, a notable exception arises in the case of the CSLDCP data sets, where the P-tuning and EFL methods are seen to be comparatively weaker than the fine-tuning approaches. The underlying reasons for this discrepancy are rooted in the fact that CSLDCP is a subject data set whose label text length is longer than that of the other data sets, making it more difficult to simplify or map to shorter sequences. As a result, P-tuning and EFL do not demonstrate the same exceptional performance. In particular, the average accuracy of the CIPLUD model is 9.19% higher than that of the fine-tuning method. Secondly, CIPLUD outperformed several representative prompt learning methods on all the datasets, with a maximum improvement in an average accuracy of 5.7% over the EFL method. This can be attributed to the fact that the CIPLUD model can use its special prompt learning design to capture and encode the semantic information of the few-shot text classification tasks. This allows it to better capture the underlying structure and semantic relationships of the data, resulting in improved performance. Finally, the average accuracy of the CIPLUD model is improved by 1.3% compared to the PET method, which also uses unlabeled data. This enhanced accuracy is a result of CIPLUD's ability to leverage unlabeled data to better choose high-confidence samples, resulting in improved performance.

In conclusion, the results show that the proposed CIPLUD model has superior performance in adapting to real-world few-shot text classification tasks. This highlights the potential of utilizing PLM information, prompt learning, and unlabeled data to improve the performance of few-shot text classification models.

### *5.2. Ablation Experiment*

CIPLUD consists of two modules: the Multiple Masks Optimization-based Prompt Learning (MMOPL) module and the One-Class Support Vector Machine-based Unlabeled Data Leveraging (OCSVM-UDL) module. We conducted a detailed analysis of the effect of the different varying components of the CIPLUD model. We further report the ablation experiments on the four FSTC tasks (EPRSTMT, CSLDCP, TNEWS, and IFLYTEK), as shown in Table 5.

**Table 5.** The comparison results between our proposed method with or without MMOPL and OCSVM-UDL.

| Method | EPRSTMT (Acc. %) | CSLDCP (Acc. %) | TNEWS (Acc. %) | IFLYTEK (Acc. %) |
|---|---|---|---|---|
| w/o MMOPL | 77.8 | 56.7 | 52.0 | 50.6 |
| w/o OCSVM-UDL | 82.1 | 59.8 | 56.4 | 52.2 |
| CIPLUD | 85.4 | 60.4 | 57.2 | 52.8 |

For "w/o MMOPL", we directly removed the multiple masks of the prompt template and replaced two masks; a single mask is meaningless for Chinese labels; for "w/o OCSVM-UDL", we removed the iterative training process using unlabeled data. Our results indicate that our approach degrades in performance without every module in most settings. This means that when any of the modules are removed, our approach does not perform as well as when all the modules are present. This indicates that individual modules are highly interdependent, and the performance of our approach relies on the presence of all the modules. Then, we found that deleting the MMOPL greatly decreased the classifier performance, and deleting the OCSVM-UDL slightly reduced the classifier performance. This is likely since MMOPL can capture more complex relationships between the prompt information and label token, while OCSVM-UDL is limited to increasing the scale of trainable data. By deleting the MMOPL, the classifier is not able to take advantage of its more complex algorithm, resulting in a decrease in performance. On the other hand, the OCSVM-UDL algorithm is limited to increasing the scale of trainable data, so deleting it will only result in a slight decrease in performance. In addition, we found that OCSVM-UDL classifiers have poor performance growth on data sets with more tag types (CSLDCP and IFLYTEK). This may be the result of the OCSVM-UDL module having difficulty obtaining uniform sampling when there are more than 50 label categories. This causes the number of recalled pseudo-labels to be imbalanced, resulting in a decrease in the accuracy of the prediction results. Overall, the results of this experiment show the importance of the MMOPL and OCSVM-UDL modules in our approach.

### 5.3. Impact of Hyper-Parameter on Constraint Boundary

To study the effect of the compactness of the constraint boundary on the model performance, we selected different values of the μ parameter in the range of [0.05, 0.95], with a step size of 0.05. On the four tasks, the CIPLUD model was observed to perform under different parameters, as shown in Figure 3. The results of the four datasets reveal that the best performance was achieved when the value of μ was in the range of [0.6, 0.8]. Extreme values of μ, either too small or too large, produced negative effects on the model's performance. A low μ resulted in a constraint that was too strict, leading to a shortage of possible pseudo-labels, while a high μ led to an overly loose constraint, which could cause overlapping boundary issues and reduce the number of feasible pseudo-labels. The research indicates that the tightness of the constraint boundary plays a substantial role in the model's performance. Hence, it is crucial to find an ideal μ that strikes a balance within the limits of the constraint boundary.

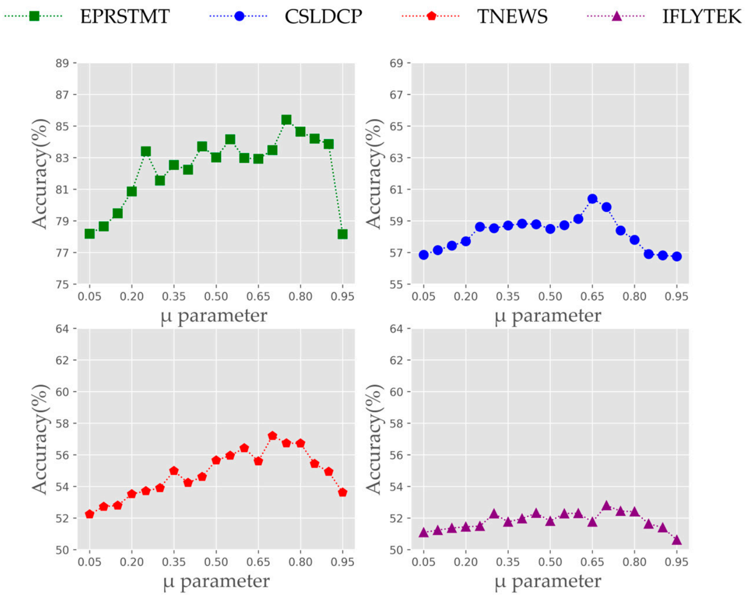

**Figure 3.** The influence of different *μ* parameters on the performance of our proposed model, CIPLUD.

In our experiments, the optimal value of μ fluctuated frequently. As a result, we have not discussed the impact of specific values. Instead, we have provided a rough estimation interval for the optimal value of μ, which can be used as a reference for future research. Taking into account the results of our investigation, we determined what value is best for each task according to the findings.

## 6. Conclusions and Future Work

In this study, we propose a CIPLUD model for Chinese few-shot text classification that utilizes improved prompt learning and unlabeled data. The model consists of two modules, the Multiple Masks Optimization-based Prompt Learning (MMOPL) module and the One-Class Support Vector Machine-based Unlabeled Data Leveraging (OCSVM-UDL) module. The MMOPL module designs universal prompt templates with multiple masks for different tasks and optimizes the predicted label of the model using joint probability and length constraints. The OCSVM-UDL module assigns pseudo-labels to the unlabeled data through a one-class support vector machine model and filters noise data from the unlabeled data. The new training data are created by blending the pseudo-labeled data with the few-shot-labeled data, and the process is repeated until the performance of the text classification model stabilizes. The performance of the CIPLUD model has been effective, as evidenced by the compared experimental results on the Chinese FSTC tasks. Moreover, we also designed the experiments to justify the component setting's rationality in CIPLUD.

For future work, we plan to refine the proposed model in several directions. As we did find when the number of label categories in a task increases, the candidate pseudo-labels obtained through semi-supervised training may be affected by imbalanced label categories. It is expected that the CIPLUD model could be further improved with a more uniform pseudo-label sampling method. We will further explore the potential of prompt learning in Chinese FSTC tasks and investigate the effectiveness of combining prompt learning with semi-supervised learning. We hope that our findings will inspire future research in the prompt learning area and contribute to the advancement of prompt learning for Chinese few-shot text classification tasks.

**Author Contributions:** Funding acquisition, Z.C.; Project administration, T.H.; Validation, Z.Y. and J.X.; Writing—original draft, T.H.; Writing—review and editing, J.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Social Science Foundation Western Project of China under NO. 19XTQ010.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
2. Chen, W.; Xu, Z.; Zheng, X.; Yu, Q.; Luo, Y. Research on Sentiment Classification of Online Travel Review Text. *Appl. Sci.* **2020**, *10*, 5275. [CrossRef]
3. Xu, G.; Liu, P.; Zhu, Z.; Liu, J.; Xu, F. Attention-Enhanced Graph Convolutional Networks for Aspect-Based Sentiment Classification with Multi-Head Attention. *Appl. Sci.* **2021**, *11*, 3640. [CrossRef]
4. Wang, Y.; Guo, J.; Yuan, C.; Li, B. Sentiment Analysis of Twitter Data. *Appl. Sci.* **2022**, *12*, 11775. [CrossRef]
5. Eminagaoglu, M. A new similarity measure for vector space models in text classification and information retrieval. *J. Inf. Sci.* **2022**, *48*, 463–476. [CrossRef]
6. Khan, S.U.R.; Islam, M.A.; Aleem, M.; Iqbal, M.A. Temporal specificity-based text classification for information retrieval. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 2915–2926. [CrossRef]

7. Ebadi, N.; Jozani, M.; Choo, K.K.R.; Rad, P. A memory network information retrieval model for identification of news misinformation. *IEEE Trans. Big Data* **2021**, *8*, 1358–1370. [CrossRef]
8. Duan, K.; Du, S.; Zhang, Y.; Lin, Y.; Wu, H.; Zhang, Q. Enhancement of Question Answering System Accuracy via Transfer Learning and BERT. *Appl. Sci.* **2022**, *12*, 11522. [CrossRef]
9. Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le, Q.V. Finetuned Language Models are Zero-Shot Learners. *arXiv* **2021**, arXiv:2109.01652.
10. Zhong, R.; Lee, K.; Zhang, Z.; Klein, D. Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 2856–2878.
11. 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, Online, 6–11 June 2021; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 5203–5212.
12. 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.
13. Schölkopf, B.; Williamson, R.C.; Smola, A.; Shawe-Taylor, J.; Platt, J.C. Support vector method for novelty detection. *Adv. Neural Inf. Process. Syst.* **2000**, *12*, 582–588.
14. Xu, L.; Lu, X.; Yuan, C.; Zhang, X.; Xu, H.; Yuan, H.; Wei, G.; Pan, X.; Tian, X.; Qin, L.; et al. Fewclue: A chinese few-shot learning evaluation benchmark. *arXiv* **2021**, arXiv:2107.07498.
15. Yan, L.; Zheng, Y.; Cao, J. Few-shot learning for short text classification. *Multimed. Tools. Appl.* **2018**, *77*, 29799–29810. [CrossRef]
16. Xu, J.; Du, Q. Learning transferable features in meta-learning for few-shot text classification. *Pattern. Recogn. Lett.* **2020**, *135*, 271–278. [CrossRef]
17. Pang, N.; Zhao, X.; Wang, W.; Xiao, W.; Guo, D. Few-shot text classification by leveraging bi-directional attention and cross-class knowledge. *Sci. China. Inform. Sci.* **2021**, *64*, 130103. [CrossRef]
18. Wang, D.; Wang, Z.; Cheng, L.; Zhang, W. Few-Shot Text Classification with Global–Local Feature Information. *Sensors* **2022**, *22*, 4420. [CrossRef]
19. Pan, C.; Huang, J.; Gong, J.; Yuan, X. Few-shot transfer learning for text classification with lightweight word embedding based models. *IEEE Access* **2019**, *7*, 53296–53304. [CrossRef]
20. Zheng, J.; Cai, F.; Chen, W.; Lei, W.; Chen, H. Taxonomy-aware learning for few-shot event detection. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 3546–3557.
21. Huisman, M.; van Rijn, J.N.; Plaat, A. A survey of deep meta-learning. *Artif. Intell. Rev.* **2021**, *54*, 4483–4541. [CrossRef]
22. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015.
23. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3630–3638.
24. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4077–4087.
25. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
26. Zhang, N.; Li, L.; Chen, X.; Deng, S.; Bi, Z.; Tan, C.; Huang, F.; Chen, H. Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021.
27. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
28. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.
29. 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.
30. 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. *arXiv* **2021**, arXiv:2107.13586. [CrossRef]
31. Schick, T.; Schütze, H. 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, Online, 6–11 June 2021; pp. 2339–2352.
32. 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, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4980–4991.
33. Wang, S.; Fang, H.; Khabsa, M.; Mao, H.; Ma, H. Entailment as few-shot learner. *arXiv* **2021**, arXiv:2104.14690.
34. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *arXiv* **2021**, arXiv:2103.10385.

35. Jiang, Z.; Xu, F.F.; Araki, J.; Neubig, G. How can we know what language models know? *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 423–438. [CrossRef]
36. 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 November 2021; pp. 3045–3059.
37. Vu, T.; Barua, A.; Lester, B.; Cer, D.; Iyyer, M.; Constant, N. Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 9279–9300.
38. Domingues, R.; Filippone, M.; Michiardi, P. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern. Recogn.* **2018**, *74*, 406–421. [CrossRef]
39. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef]
40. Fei, G.; Liu, B. Breaking the Closed World Assumption in Text Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 506–514.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.