applied sciences

MDPI

Article

Investigating Prompt Learning for Chinese Few-Shot Text Classification with Pre-Trained Language Models

Chengyu Song [1,†], Taihua Shao [1,*], Kejing Lin [2,†], Dengfeng Liu [1], Siyuan Wang [1] and Honghui Chen [1]

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology, No. 109 Deya Street, Changsha 410073, China
[2] School of Information Resources Management, Renmin University of China, No. 59 Zhongguancun Street, Beijing 100872, China
* Correspondence: shaotaihua13@nudt.edu.cn
† These authors contributed equally to this work.

Abstract: Text classification aims to assign predefined labels to unlabeled sentences, which tend to struggle in real-world applications when only a few annotated samples are available. Previous works generally focus on using the paradigm of meta-learning to overcome the classification difficulties brought by insufficient data, where a set of auxiliary tasks is given. Accordingly, prompt-based approaches are proposed to deal with the low-resource issue. However, existing prompt-based methods mainly focus on English tasks, which generally apply English pretrained language models that can not directly adapt to Chinese tasks due to structural and grammatical differences. Thus, we propose a prompt-based Chinese text classification framework that uses generated natural language sequences as hints, which can alleviate the classification bottleneck well in low-resource scenarios. In detail, we first design a prompt-based fine-tuning together with a novel pipeline for automating prompt generation in Chinese. Then, we propose a refined strategy for dynamically and selectively incorporating demonstrations into each context. We present a systematic evaluation for analyzing few-shot performance on a wide range of Chinese text classification tasks. Our approach makes few assumptions about task resources and expertise and therefore constitutes a powerful, task-independent approach for few-shot learning.

Keywords: few-shot learning; prompt learning; template generation; demonstration learning

Citation: Song, C.; Shao, T.; Lin, K.; Liu, D.; Wang, S.; Chen, H. Investigating Prompt Learning for Chinese Few-Shot Text Classification with Pre-Trained Language Models. *Appl. Sci.* **2022**, *12*, 11117. https://doi.org/10.3390/app122111117

Academic Editor: Rafael Valencia-Garcia

Received: 8 September 2022
Accepted: 31 October 2022
Published: 2 November 2022

1. Introduction

Text classification (TC) is a key task in natural language processing (NLP), which aims to assign predefined labels or classes to input texts [1]. TC has been widely applied in many real-world applications such as social media analysis [2,3], question answering [4], and information retrieval [5], etc. However, in real-world applications, a major problem of TC is the insufficient human-annotated data. Thus, few-shot TC has been proposed to solve the low-resource problem by limiting the amount of annotated data. Additionally, research on low-resource languages [6–10] such as Chinese, Korean, Spanish, etc., is yet to be fully explored.

Meta-learning is one of the most successful techniques in the practice of few-shot learning [11–13] , which learns the meta knowledge from the support classes and then generalizes it to other unseen classes. However, the generalization ability of meta-learning-based approaches mainly relies on abundantly seen classes that can not be easily collected. Therefore, prompt learning is proposed to alleviate this issue, which provides natural language hints and transforms the downstream tasks into masked language modeling problems. We show the main differences between the prompt-based approach and the previous training methods in Figure 1. Thus, the prompt-based methods can quickly adapt to new tasks with limited annotated data and reach the true few-shot setting [14], i.e., identically small training and validation sets.

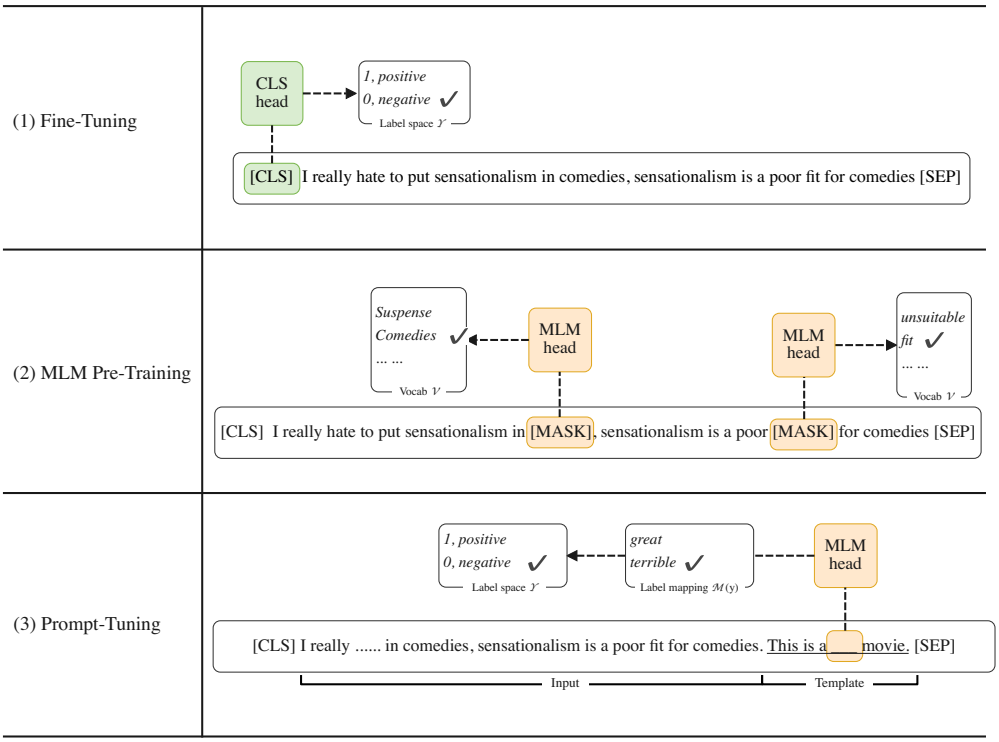

Figure 1. The example of fine-tuning, masked language modeling, and prompt-tuning. (1) The Fine-Tuning approach uses the representation of the headers special token as the representation of the whole sentence for prediction, (2) the masked language modeling (MLM) obtains the sentence representation by masking some words for the model to predict, and (3) the prompt-based fine-tuning (prompt-tuning) method allows the model to predict the textual answers directly by adding a sequence with a mask as a hint.

Intuitively, the manually-designed prompt is the easiest way to elicit semantic knowledge from the language models [15,16]. Yet, it is possible that manually created prompts are sub-optimal [17] as well as labor-intensive. In this light, prompt engineering that focuses on generating prompts automatically has been widely explored. While it is possible to obtain high-performance prompt-learning models for few-shot tasks in English, there are still many others that, due to lack of resources or attention, have not yet benefited from advances in the field of prompts. Despite the promising achievements, most existing methods only consider generating English prompts and Chinese prompt engineering methods are yet to be explored.

Further, one of the best-performing training techniques is demonstration learning [18,19], which concatenates the query with one selected example from each category for fine-tuning. Existing demonstration learning methods typically select the example at random or on the basis of similarity. However, we argue the previous demonstration learning methods are not guaranteed to prioritize the most informative example in the absence of a proper validation mechanism.

In this paper, we propose a prompt-based Chinese text classification framework to solve the classification bottleneck in the true few-shot setting, i.e., a small number of training and validation samples, along with moderately-sized language models. This framework mainly consists of two novel parts, namely, the template generation module and the demonstration filtering module. In detail, we introduce an automatic prompt generation process, including a pruned brute-force search to identify the best working templates that allow us to cheaply obtain effective prompts that match or outperform our manually chosen ones. In addition, we adopt the idea of incorporating demonstrations as additional context and present an advanced candidate filtering method using mutual information and cosine similarity. Consequently, this joint correlation scoring function

allows the model to train with more valuable examples than random selection. Experiments on a set of Chinese text classification tasks under true few-shot learning settings show that our proposal achieves notable improvements over strong few-shot learning baselines.

The main contributions in this paper can be summarized as follows:

- To the best of our knowledge, we are the first to apply prompt learning to few-shot TC as well as to design the task-agnostic template generation strategies and label representation in the Chinese domain.
- We design a joint correlation scoring function to be capable of selecting the most related examples for fine-tuning so as to raise the classification performance.
- We evaluate our proposal against the strong baselines on a set of Chinese text classification tasks under a true few-shot setting. The experimental results demonstrate the advantage of our proposal.

2. Related Work

In this section, we first review related works about TC from two aspects, i.e., the data-rich setting and the few-shot setting. Moreover, we summarize the previous works on prompt learning including diverse methods of prompt engineering.

2.1. Text Classification

As a fundamental task in natural language processing (NLP), text classification has attracted lots of research interest in the last decade. Typical text classification models rely on various well-designed neural networks to learn the semantic patterns based on abundantly annotated data [20–24]. Instead, few-shot text classification methods generally adopt metric-based [25–29] meta-learning to predict the labels of the inputs.

2.1.1. Data-Rich Text Classification

When a large number of human-annotated data are available, TC models can easily converge. With the development of deep learning, various types of neural networks have been used for TC. For example, ref. [30] propose a recurrent architecture based on an LSTM network trained using discriminative fine-tuning for text classification. In addition, ref. [31] presents a comparatively simple CNN-based model with one layer convolution structure that is placed on top of word embeddings. Moreover, ref. [32] rely on the application of attention both at the word level and the sentence level.

However, these models generally require quantities of manually-annotated data to realize the model convergence, which is infeasible and requires high labor costs.

2.1.2. Few-Shot Text Classification

In recent years, numerous research interest has been addressed in solving text classification in the few-shot scenarios using metric-based methods. For example, ref. [33] propose a Siamese neural network for document classification, along with a well-designed contrastive loss. Moreover, ref. [34] propose a hierarchical attention prototypical network that utilizes the attention layer cross feature level, word level and instance level. Further, ref. [35] introduce two regularization matching losses to improve the model performance.

Although meta-learning-based approaches create a data-in-sufficient scenario for each meta-tasks, they still require many held-out examples for training and evaluation altogether.

2.1.3. Pre-Trained Language Models

In both data-rich and few-shot scenarios, pretrained language models play an important role as the backbone of various text classification models. For example, masked language models [36,37] aim to predict masked text pieces based on surrounded context. Left-to-right language models [38] are a variety of auto-regressive language models that predict the upcoming words or assign a probability to a sequence of words. Prefix language models [39,40] use a left-to-right language model to decode the answer conditioned on a

separate encoder for inputs with fully-connected mask, i.e., the parameters of the encoder and decoder are not shared.

Benefiting from these powerful pretrained language models, the text classification models can quickly adapt to downstream tasks with only a few annotated data.

2.2. Prompt Learning

Prompt learning methods are fueled by the birth of GPT-3 [19] and have outstanding performance in widespread NLP tasks.

2.2.1. Template Engineering

Template engineering is the process of creating a sequence of tokens that improves the model performance for the downstream tasks, which plays an important role in prompt learning. Existing template engineering methods can be roughly classified into two categories, namely, manual template engineering and automated template learning. Hand-crafted templates are the most natural and intuitive way. For example, ref. [41] provide manually created cloze-style templates to probe features in language models. Moreover, ref. [15] design a set of templates for various NLP tasks and achieve notable performance gains.

While hand-crafted templates do allow for solving various tasks with some degree of accuracy improvements, they generally require time and experience, hence sometimes failing to manually discover the most optimal templates [42]. To address this problem, several methods propose to automate the template design process from two perspectives, i.e., discrete prompts [18,43,44] and continuous prompts [45–47]. For example, ref. [18] introduce the seq2seq pretrained model T5 [40] into template search process. In addition, continuous prompts perform prompting directly in the embedding space of the model. For example, ref. [45] propose a method that prepends a sequence of continuous task-specific vectors to the input while keeping the model parameters frozen.

However, the above-mentioned template generation methods are specifically designed for English tasks and cannot directly be adapted to the Chinese domain since the large gap between the two languages.

2.2.2. Demonstration Learning

The core idea of demonstration learning (e.g., prompt augmentation) is that providing a few additional answered prompts to the actual prompt can be useful for few-shot learning, where two aspects have been widely studied, namely, sample selection and sample ordering [17]. Researchers have found that different demonstration example selection methods can result in various performances. Thus, ref. [18] address this issue by calculating the similarity between query and support examples. Moreover, ref. [48] provide both positive and negative samples that highlight things to avoid. For sample order, ref. [48] find that the order of demonstration examples provided to the model plays an important role in improving the model performance. In addition, ref. [49] search for good training sample permutations as augmented prompts and learn a separator token between the prompts to obtain improvements.

However, existing methods of prompt augmentation barely consider evaluating how informative are the examples to the query. This consideration is necessary as high performance can be achieved by providing informative demonstration examples.

3. Approach

In this section, we first introduce the task formulation of the few-shot text classification in Section 3.1. Then, in Section 3.2, we explore the principled ways of automatically generating templates in the Chinese domain. Finally, we propose a refined demonstration strategy in Section 3.3, which improves the model performance by selecting highly related examples for training. We plot the workflow of our proposal in Figure 2.

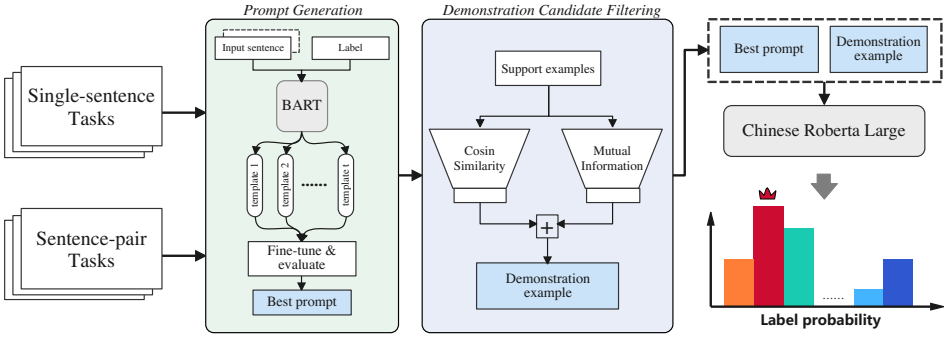

Figure 2. The workflow of the proposed model.The green rectangle represents the prompt generation module in Section 3.2 and the blue rectangle represents the demonstration candidate filtering in Section 3.3.

3.1. Task Formulation

3.1.1. Few-Shot Text Classification

Essentially, text classification is a kind of sequence labeling task. Specifically, given any input sentence x from a training set $\mathcal{D}_{train}$ and a predefined label set Y, the classification model f_e can return the label y based on the text features, i.e., $f_e(x) \rightarrow y, y \in Y$. Typically, the success of f_e mapping to the test set $\mathcal{D}_{train}$ is attributed to quantities of training examples, yet this becomes unacceptable in practical applications.

Hence, in this paper, we aim at investigating the model performance of text classification in true few-shot scenarios. Formally, for each category in $\mathcal{D}_{train}$, only K samples are given, such that the total number of training examples is $|\mathcal{D}_{train}| = K \times |Y|$. Traditionally, using a development set with a large number of instances leads to better performance [18], yet this does not fit our few-shot claim. Therefore, we also limit the number of examples in the development set to the same as the training set, i.e., $|\mathcal{D}_{train}| = |\mathcal{D}_{dev}|$.

3.1.2. Prompt-Based Fine-Tuning

When performing standard fine-tuning, previous methods usually train a task-specific head f_c that maximizes the log probability of the correct label through a $softmax$ function as:

$$\begin{aligned} p(y_i|x_i) &= p(\hat{y} = y_i \mid x_i) \\ &= softmax(W_y \cdot f_c(h_{[CLS]})) \\ &= \frac{exp(W_{y_i} \cdot f_c(h_{[CLS]}))}{\sum_{y' \in Y} exp(W_{y'} \cdot f_c(h_{[CLS]}))}, \end{aligned} \tag{1}$$

where $h_{[CLS]}$ is the hidden vector of the special head token $[CLS]$, and $W \in \mathbb{R}^{|y| \times d}$ is a set of randomly initialized parameters introduced before fine-tuning.

Contrarily, when performing prompt-based fine-tuning, language model $\mathcal{L}$ is directly tasked with "auto-completing" natural language prompts. It usually takes two steps to form a prompt, namely, label representing $\mathcal{M}()$ and template transformation $\mathcal{T}()$. Specifically, $\mathcal{M}()$ converts the label space into several individual words in the vocabulary. On the other hand, $\mathcal{T}()$ is a transformation function that concatenates the input x_i with a predefined template that contains a $[MASK]$ position. The probability of predicting the label $y_i \in Y$ is then transformed to:

$$\begin{aligned} p(y_i|x_i) &= p([MASK] = \mathcal{M}(y_i) \mid \mathcal{T}(x_i)) \\ &= softmax(W_{\mathcal{M}(y_i)} \cdot h_{[MASK]}) \\ &= \frac{exp(W_{\mathcal{M}(y_i)} \cdot h_{[MASK]})}{\sum_{y' \in Y} exp(W_{\mathcal{M}(y')} \cdot h_{[MASK]})}, \end{aligned} \tag{2}$$

where $W_{\mathcal{M}(y)}, h_{[MASK]}$ denote the hidden vector of $[MASK]$ and the label word $\mathcal{M}(y)$, respectively.Compared with standard fine-tuning, prompt-based fine-tuning re-uses the pretrained weights and does not introduce any new parameters. Moreover, it also reduces the gap between pretraining and fine-tuning, formulating an effective few-shot scenario.

Note that in our implementation, the label words used for label representation function $\mathcal{M}()$ are manually-designed. Since the text classification tasks contain a small number of categories and are all easily expressible. Therefore manual design is the easiest and most efficient way of designing label words.

3.2. Template Generation

Next, we zoom in on generating a diverse set of templates, which are supposed to work well for all examples in $\mathcal{D}_{train}$. To address this challenging problem, we propose to adopt the core idea of Chinese BART [50], a large pretrained denoising sequence-to-sequence model with a standard transformer architecture (https://huggingface.co/fnlp/bart-base-chinese, accessed on 30 October 2022).

BART is trained by corrupting documents and then optimizing a reconstruction loss, i.e., the cross-entropy between the decoder's output and the original document. One of the pretraining tasks of BART is text infilling, in which several text spans are sampled and replaced with a single $[MASK]$ token. For example, given the input "This shirt uses a fresh blue tone to show the $[MASK]$ of women's hearts.", BART is trained to generate "beauty and purity" for the replacement for the $[MASK]$ position. Accordingly, BART is well suited for generating templates since the original input can be treated as predefined constraints. Hence, the filled blanks plus the input sentence can easily form the template $\mathcal{T}$. Formally, given an input tuple $(x_i, y_i) \in \mathcal{D}_{train}$, the BART prompt format can be returned considering the following permutations:

$$\mathcal{T}_{BART}(x_i, y_i) = \left\{ \begin{array}{l} x_i[MASK]y_i, \\ y_i[MASK]x_i, \end{array} \right\}, \tag{3}$$

where the masked positions depend on the BART model to fill.

To effectively obtain a large set of diverse templates, we follow [18] to apply beam search in the decoding process to obtain multiple template candidates. At each position, instead of choosing the token with the highest probability and generate a sequence subsequently as follows:

$$\mathcal{T}_c = \underset{\forall t^{(k)} \in \mathcal{V}}{Beam\text{-}search} \sum_{k=1}^{L^j_{T5}} \sum_{(x_i, y_i) \in \mathcal{D}_{train}} logP_{BART}(t^{(k)} | t^{(k-1)}, \cdots, t^{(1)}, \mathcal{T}^j_{T5}(x_i, y_i)), \tag{4}$$

where $\mathcal{V}$ denotes the vocabulary of the pretrained language model, $(t^{(k)}, \cdots, t^{(1)})$ are the tokens in a template. By using Equation (4), we obtain a list of candidate templates. With these diverse generated prompts, we perform a prompt fine-tuning on $\mathcal{D}_{dev}$ to pick the best performing template $\hat{\mathcal{T}}$ as follows:

$$\underset{[mask] \in \mathcal{V}}{MAX} \left\{ \sum_{(x_i, y_i) \in \mathcal{D}_{train}, \mathcal{D}_{dev}} logP_{\mathcal{L}}([MASK] = \mathcal{M}(y_i) | \mathcal{T}^j_c(x_i)) \right\}, \mathcal{T}^j_c \in \mathcal{T}_c, \tag{5}$$

where $P_{\mathcal{L}}$ is the predicted probability of the language model. The small size of $\mathcal{D}_{train}$ and $\mathcal{D}_{dev}$ ensure the computation and time complexity of our proposed fine-tuning process remain acceptable.

3.3. Demonstration Candidate Filtering

We design a demonstration candidate filtering strategy by utilizing mutual information (MI), which can select the most informative example from the training set.

3.3.1. Training Examples as Demonstrations

Despite the enormous parameters contained in GPT-3, its excellent few-shot learning ability is largely attributed to a simple strategy that concatenates the input with examples randomly drawn from the training set. This simple strategy has been leveraged for fine-tuning the language model, also known as demonstration learning. Specifically, a set of examples (x_i, y_i) are sampled from the training set $\mathcal{D}_{train}$ and converted by a template $\mathcal{T}()$ and a label mapping function $\mathcal{M}(y)$. Here, we denote the combination of these two prompt conversions as $\tilde{\mathcal{P}}()$:

$$\tilde{\mathcal{P}}(x_i, y_i) = \begin{cases} y_i \ \ \rightarrow \mathcal{M}(y_i) \\ x_i \rightarrow \mathcal{T}(x_i) \end{cases} \quad (x_i, y_i) \in \mathcal{D}_{train}. \tag{6}$$

Then, the selected examples are concatenated in the order of their labels with support examples $x_{in}^{c_i}$ as:

$$\mathcal{T}(x_q) \oplus \tilde{\mathcal{P}}(x_{in}^{(c_1)}, y^{(c_1)}) \oplus \tilde{\mathcal{P}}(x_{in}^{(c_2)}, y^{(c_2)}) \oplus \cdots \oplus \tilde{\mathcal{P}}(x_{in}^{(c_m)}, y^{(c_m)}), \tag{7}$$

where $\oplus$ denotes the concatenation of sequences, and c_m is the total number of demonstration examples.

3.3.2. Joint Correlation Score Function

The superiority of demonstration learning relies on the ability to demonstrate how the language model should provide the answer to the actual prompt instantiated with the input. For example, providing a prompt of "China's capital is $[MASK]$" prefaced by a few examples such as "Great Britain's capital is London. Japan's capital is Tokyo". These demonstrations enable strong language models to learn repetitive patterns. In this light, examples that are semantically close to the query sample in the embedding space consistently give rise to a strong performance. The existing approaches are suboptimal for few-shot text classification since they can not ensure the most similar examples are learned as well as introduce additional parameters such as SBERT [18] and increase the computation complexity.

Accordingly, we propose a joint correlation scoring function that combines the cosine similarity and the point-wise mutual information. Specifically, given a query input x_q and a support example x_i, we first feed them into an encoder to extract their sentence-level embeddings $e(x_q), e(x_i)$. Then, the cosine similarity score can be calculated by:

$$\mathcal{S}(x_q, x_i) = dist(e(x_q), e(x_i)) = \frac{e(x_q) \cdot e(x_i)}{||e(x_q)||_2^2 \cdot ||e(x_i)||_2^2}. \tag{8}$$

The point-wise mutual information using data collected by information retrieval was proposed as an unsupervised measure for the evaluation of the semantic similarity of words. It is based on word co-occurrence using counts collected over the corpora. Formally, given two words w_1 and w_2, their PMI is measured as:

$$PMI(w_1, w_2) = \log_2 \frac{p(w_1, w_2)}{p(w_1) \times p(w_2)}, \tag{9}$$

which indicates the degree of statistical dependence between w_1 and w_2, and it can be used as a measure of the semantic similarity of the two words. Then, the mutual information between two sentences, S_1 and S_2 can be calculated as the sum of PMI between their contained words as follows:

$$I(S_1, S_2) = \sum_{w_i \in S_1} \sum_{w_j \in S_2} PMI(w_i, w_j). \tag{10}$$

Recall that in the situation of prompt learning, a support set is typically formulated as $\mathcal{D}_{support} = (x_i, y_i)$, where x_i and y_i denote the sentence and label of the ith training sample. Considering the similarity and the mutual information calculation illustrated above, the demonstration example of each query is selected by a joint correlation scoring function as:

$$\underset{x_i \in \mathcal{D}_{support}}{MAX} \left\{ Score(x_q, x_i) = I(x_q, x_i) + \mathcal{S}(x_q, x_i) \right\}, \tag{11}$$

We denote the concatenation of the query and the selected demonstration example as a context c_m:

$$c_m = \mathcal{T}(x_q) \oplus \tilde{\mathcal{T}}(x_i) \tag{12}$$

For each context c_m, we fine-tune it on $\mathcal{D}_{train}$ and $\mathcal{D}_{dev}$ to pick the best performing candidate. With the filtered support set $\mathcal{D}_{support}$, we then perform a standard prompt-based training procedure following Equation (2).

We summarize the entire training procedure of our proposal in Algorithm 1. For each input tuple in the training set, we first replace the label with predefined label words in line 1. Then, we insert masked positions and special tokens into the gaps between input and labels in lines 3 to 4. After that, we employ BART to fill in the [MASK] and perform the tine-tuning on each generated candidate so as to select the best-performing template in lines 5 to 6. Next, for each query, we transform them to the prompt format in line 9 and calculate the correlation score between the query and every support example in lines 11 to 13. Finally, we chose the example with the largest correlation score for demonstration learning in line 14.

Algorithm 1: Demonstration filtering based on joint correlation scoring

Input: $\mathcal{D}_{train} = \{x_i, y_i\}, where\ i = 1, 2, \cdots, n$: The training set;
q: the query sentences contained in $\mathcal{D}_{train}$; $\mathcal{M}()$: label representation function;
Output: $x_{demonstration}$: The best-suited demonstration example for each query.

1 **foreach** $\{x_i, y_i\} \in \mathcal{D}_{train}$ **do**
2 label representation: $y_i \rightarrow \mathcal{M}(y_i)$
3 add placeholders: $[MASK] \xrightarrow{insert} \{x_i, y_i\}$
4 add special tokens: $\mathcal{T}_{BART} = \{[CLS], x_i, [MASK], y_i, [SEP]\}$
5 fill in placeholders: $\mathcal{T}_{BART} \xrightarrow[generation]{Beam-search} \mathcal{T}_c$
6 filtering: $\hat{\mathcal{T}} \leftarrow \underset{\mathcal{T}_i \in \mathcal{T}_c}{argmax} P_{\mathcal{L}}(y | x_q \oplus \mathcal{T}), \mathcal{T} \in \mathcal{T}_c$
7 **foreach** q *in* $\mathcal{D}_{train}$ **do**
8 **for** $\{x_{in}^1, \cdots, x_{in}^m\} \in \mathcal{D}_{similar}$ **do**
9 transform input to prompt: $y \rightarrow \mathcal{M}(y)\ \ (x_{in}^i, \mathcal{M}(y_i), t_i) \rightarrow \tilde{\mathcal{T}}(x_{in}, y_i)$
10 **foreach** $x_i \in \mathcal{D}_{support}$ **do**
11 calculate cosin similarity: $S(x_q, x_i) = dist(e(x_q), e(x_i))$
12 calculate mutual information: $I(S_1, S_2) = \sum_{w_i \in S_1} \sum_{w_j \in S_2} PMI(w_i, w_j)$
13 calculate correlation score: $Score(x_q, x_i) = I(x_q, x_i) + \mathcal{S}(x_q, x_i)$
14 filtering: $\underset{x_i \in \mathcal{D}_{support}}{MAX} \left\{ Score(x_q, x_i) = I(x_q, x_i) + \mathcal{S}(x_q, x_i) \right\}$
15 **return**: $x_{demonstration}$

4. Experiments

4.1. Datasets

We evaluate the performance of our proposal on CLUE (https://github.com/ChineseGLUE/ChineseGLUE, accessed on 30 October 2022), a Chinese language understanding evaluation benchmark, which consists of 9 natural language understanding tasks and a linguistically motivated

diagnostic dataset [51]. Here, we select several text classification tasks contained in CLUE for evaluation, both single-sentence and sentence-pair tasks are included, namely, AFQMC, OCNLI, TNEWS, INEWS, and BQ. We introduce the above-mentioned datasets below:

- AFQMC stands for the Ant Financial Question Matching Corpus, which comes from the Ant Technology Exploration Conference Developer competition. It is a binary classification task that aims to predict whether two sentences are semantically similar.
- OCNLI stands for Original Chinese Natural Language Inference, which is collected by closely following the procedures of MNLI. OCNLI is composed of 56 K inference pairs from five genres, namely, news, government, fiction, TV transcripts, and Telephone transcripts.
- TNEWS stands for TouTiao text classification for news titles, which consists of Chinese news published by TouTiao before May 2018, with a total of 73,360 titles. Each title is labeled with one of 15 news categories and the task is to predict which category the title belongs to.
- INEWS stands for Sentiment Analysis for Internet News, which consists of more than 7356 annotated samples. Each input sentence is labeled with one of 3 emotion labels.
- BQ refers to Question Machine for Customer Service, which is an automated question and answer systems corpus that contains 120,000 sentence pairs. The task is to predict whether one sentence is semantically similar to the other one.

The data statistics are shown in Table 1.

Table 1. Statistics of text classification tasks. We set the training sample size K of each category to 16, as well as the evaluation sample size and select 1000 samples for each category for testing.

Dataset	# Categories	# Samples	# Task
AFQMC	2	42,511	Question matching
OCNLI	3	56,000	Natural language inference
TNEWS	15	380,000	News classification
INEWS	3	7356	Sentiment analysis
BQ	2	120,000	Natural language inference

4.2. Model Summary

For all discussed models, we employ the same text encoder, i.e., the RoBERTa-wwm-large [52] to fairly compare their performance. Here, we list a series of strong baselines for comparison with our proposal in this paper.

- Fine-tune [52] is a Chinese pretrained language model that adopts a new masking strategy called whole word masking;
- PET [15] 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.
- P-tuning [45] propose to learn continuous prompts by inserting trainable variables into the embedded input.
- LM-BFF [18] uses T5 [40] to generate discrete templates automatically, and further fine-tunes the language model with a vanilla demonstration learning method.

We summarize the differences between each method and list them in Table 2. All discussed models are different in prompt designing, label words selection, and parameter updating, which allows us to make a full assessment of our proposal. Note that we adapt all baselines to the Chinese domain. For example, we switch the T5 used in LM-BFF for template generation to its Chinese version (https://huggingface.co/uer/t5-v1_1-base-chinese-cluecorpussmall, accessed on 30 October 2022).

Table 2. An organization of baselines.

Method	LM Params	Prompt Designing		Prompt Style	Fine-Tune Strategy
		Templates	Label Words		
Fine-tune	Tuned	—	—	—	—
PET	Tuned	Hand-craft	Hand-craft	Discrete	Model Ensemble
LM-BFF	Tuned	Auto	Auto	Discrete	Demonstration
P-tuning	Frozen	Auto	Auto	Gradient	Tuning-free
Ours	Tuned	Auto	Auto	Discrete	Demonstration

4.3. Hyper-Parameter Selection

When generating discrete templates in Section 3.2, we set the beam-search to 30, since the template tokens are to be tuned so that slight differences can be ignored. For evaluating our proposal, we adopt the scheme of training on a maximum sequence length of 512 tokens. We set the batch size to 128 and an initial learning rate of 1e-4 with the max training steps to 1000 during training. For the few-shot setting, we follow Xu et al. [53] to set the training sample size K of each category to 16, as well as the evaluation sample size.

4.4. Research Questions

We evaluate the performance of our proposal by addressing the following research questions:

(RQ 1) Can our proposal achieve better performance than the baselines for Chinese few-shot classification?
(RQ 2) Which component contributes more to improving the model performance, the template generation of the refined demonstration learning?
(RQ 3) How does our proposal perform with different lengths of the templates?

5. Results and Discussion

5.1. Overall Performance

To answer RQ1, we examine the few-shot Chinese text classification performance of our proposal and four competitive baselines on five public-available datasets. We present the results of all discussed models in a true few-shot setting with the same sample size of each category, i.e., $|K| = 16$ in Table 3 and the confusion matrix of our model on each corpus are shown in Appendix A. Generally, we can observe that all prompt-based models have a smaller margin of error than fine-tune, indicating that adding prompts tends to achieve stable performance.

Table 3. Model performance. The result of the best performing baseline and the best performer in each column is underlined and boldfaced, respectively.

Method	AFQMC	OCNLI	TNEWS	INEWS	BQ
Fine-tune	54.57 (±3.8)	52.36 (±6.5)	46.92 (±2.1)	44.45 (±2.0)	47.52 (±4.3)
PET	62.43 (±3.2)	58.16 (±4.7)	51.2 (±1.7)	49.92 (±1.5)	53.61 (±3.9)
LM-BFF	75.64 (±2.2)	73.82 (±3.7)	70.44 (±0.9)	68.71 (±0.8)	76.38 (±2.4)
P-tuning	73.59 (±2.3)	70.46 (±3.4)	69.12 (±0.7)	67.32 (±0.6)	68.52 (±1.7)
Ours	**77.21 (±2.3)**	**74.39 (±3.5)**	**71.17 (±1.0)**	**71.35 (±0.4)**	**76.88 (±2.0)**

In addition, our proposal is the best performer among all discussed models, with a noticeable accuracy improvement. For instance, our model presents an improvement of 1.57%, 0.57%, 0.73%, 2.64%, and 0.5% in terms of accuracy against the best performing baseline on AFQMC, OCNLI, TNEWS, INEWS, and BQ, respectively. These overwhelming results indicate that our proposal leads to consistent gains in a majority cross Chinese text classification tasks. Moreover, the major difference between our proposal and LM-BFF is

the demonstration strategy. Therefore, the comparison of our proposal against LM-BFF illustrates the strength of our proposed joint correlation scoring function.

Further, we find that P-tuning under-performs LM-BFF on all discussed tasks. For example, LM-BFF achieves an accuracy improvement of 2.05%, 3.36%, and 1.32% on AFQMC, OCNLI, and TNEWS against P-tuning. It can be explained by the fact that the combination of discrete prompts and demonstration learning prefers fewer inputs than continuous prompts. Regarding the template style, automatically generated templates generally outperform the hand-crafted templates on all datasets. For example, PET shows an accuracy decrease of 22.77%, 14.91%, and 23.27% on BQ against LM-BFF, P-tuning, and our proposal, which reflects that although a manual prompt is more intuitive than an automated one, it is more easily trapped in the local optimist. Moreover, automatically generated templates perform more stably than manually designed ones, meaning that automatically generated templates have more generalization capabilities.

5.2. Ablation Study

For RQ2, we perform an ablation study by comparing our proposal with its variants to analyze the effectiveness of each component. Specifically, we produce four variants for comparison, including: (1) "w/o demo" that removes the whole demonstration learning module and utilizes a fine-tuning; (2) "w/o demo (full)" that removes the joint correlation scoring function and adopts a full demonstration following LM-BFF; (3) "w/o demo (random)" that removes our proposed scoring function and employs a random sample as a demonstration example; (4) "w/o generation (man)" that removes the template generation module and uses manual-crafted templates. The results are shown in Table 4.

From Table 4, we can observe that the removal of any component in our proposal leads to a performance decrease, indicating that all components contribute to the model performance. Further, the removal of the demonstration module has the greatest impact on model performance, illustrating that providing random examples as demonstrations can help the language model to capture the answer patterns for prompts. Moreover, comparing "w/o demo (full)" and "w/o demo (random)", we can notice that "w/o demo (full)" outperforms "w/o demo (random)" in all cases. It can be explained by the fact that the random selection sometimes ignores the informative examples for the query. In addition, the comparison between "w/o demo (full)" and our proposal demonstrates that our proposed joint correlation scoring function can effectively select the informative demonstration example for each query and thus improve the model performance.

Table 4. Ablation study results of our proposal on 5 datasets. The results of the best performer in each column are bolded. The biggest drop in each column is appended with ▼.

Variants	AFQMC	OCNLI	TNEWS	INEWS	BQ
w/o demo	70.46 ▼	68.32 ▼	66.71 ▼	64.18 ▼	73.19 ▼
w/o demo (full)	76.53	72.86	70.51	69.34	74.26
w/o demo (random)	75.44	71.82	69.13	68.52	73.92
w/o generation (man)	74.53	72.95	68.49	69.02	74.37
Ours (original)	**77.21**	**74.39**	**71.17**	**71.35**	**76.88**

In addition, "w/o generation (man)" loses the performance competition to our original proposal in terms of accuracy on all tasks. We attribute the reason that the manual-crafted templates are usually sub-optimal to the model training.

5.3. Impact of Template Length

To answer RQ3, we vary the template length in $\{5, 10, 20, 30, 50\}$ and keep other settings to our default configurations. We re-examine the performance of the original and the hand-crafted version of our proposal on all tasks. The model performance under different template lengths is shown in Figure 3 and the model performance of each epoch

with a template length of 20 is shown in Figure 4. Clearly, with the increase in the template length, both versions of our proposal show a consistent pattern in the model performance, i.e., increases first to reach the top and then goes down. While the dropping tendency reflects the fact that the overlong templates inevitably add noise, making it difficult for demonstration and classification.

Interestingly, comparing the performance drop caused by the increase in the template length, the drop on TNEWS is more obvious than that on other tasks. It can be stemmed from the dataset itself, in which the category number of TNEWS is larger than other datasets.

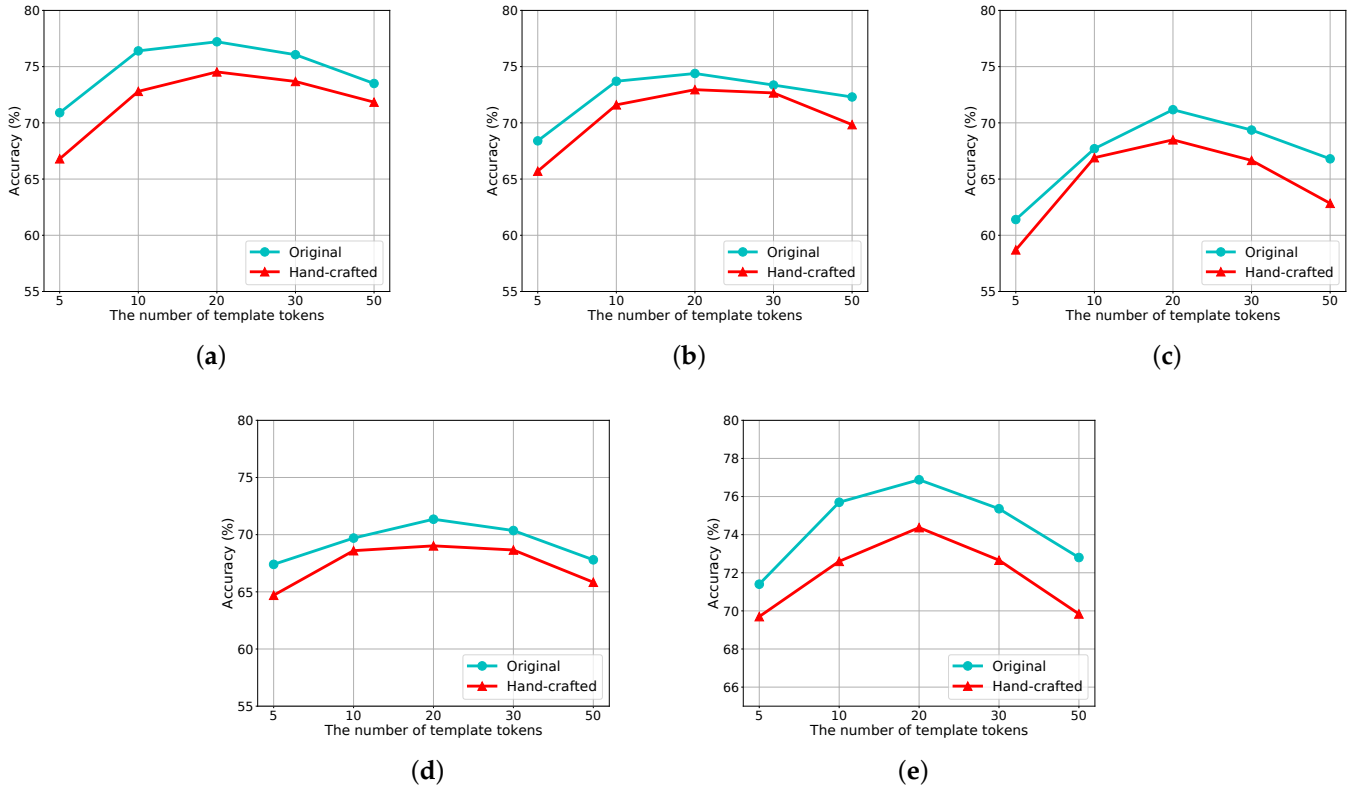

Figure 3. Model performance of TaxonPrompt and its variants with various language models on Few-Event. (**a**) Accuracy on AFQMC. (**b**) Accuracy on OCNLI. (**c**) Accuracy on TNEWS. (**d**) Accuracy on INEWS. (**e**) Accuracy on BQ dataset.

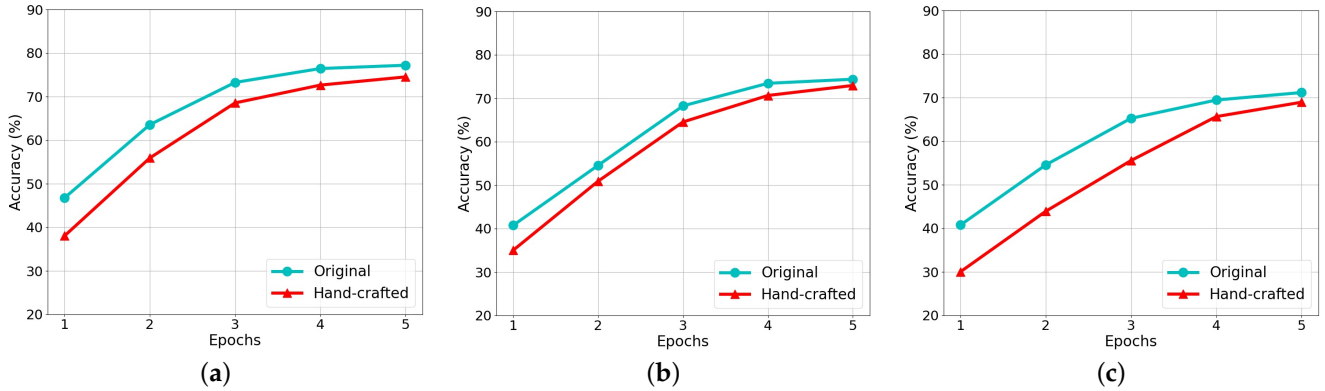

Figure 4. *Cont.*

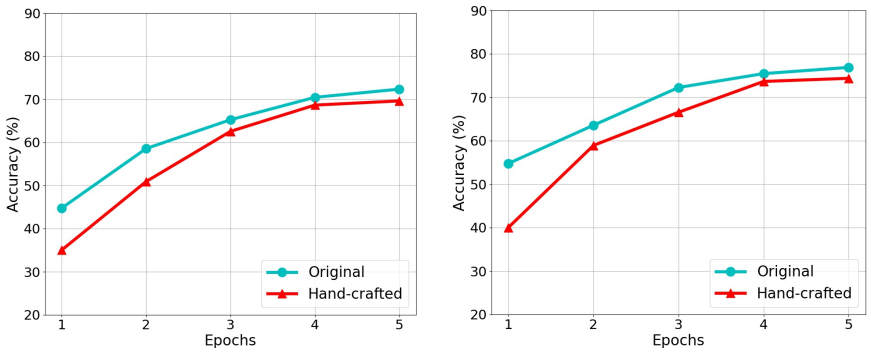

Figure 4. Model performance of our model and its variants per epoch on different corpora. (**a**) Accuracy on AFQMC. (**b**) Accuracy on OCNLI. (**c**) Accuracy on TNEWS. (**d**) Accuracy on INEWS. (**e**) Accuracy on BQ dataset.

5.4. Practical Implications and Technical Challanges

The practical implication of our work is that our proposed Chinese text classification framework shows notable improvements over comparable baselines. Our proposed template generation model is able to generate high-quality task-specific templates for each corpus. In addition, experiment results further illustrate our proposed joint correlation score function is able to select informative samples as demonstration examples.

Although our proposed framework achieves state-of-the-art performances on various independent tasks, prompt learning for Chinese text classification is yet to be fully explored. In summary, the technical difficulties and challenges can be summarized as follows:

- To achieve the best performance on different tasks, the template generation module needs to be retrained on different corpus in order to generate task-specific templates, which is inefficient in real-life applications.
- During template evaluation, the best-performing template needs to be selected by zero-shot prediction on the validation set, which is acceptable when the sample size is small; however, it can be time-consuming in traditional text classification tasks.
- In order to generate high-quality templates, text-to-text pretrained models are used for fine-tuning and text generation tasks, a process that requires a high level of computer hardware. For example, the BART we use requires at least 600 M of memory for reading the model.

6. Conclusions

We propose a prompt learning framework for Chinese few-shot text classification. Our proposal utilizes a template generation module specially designed for Chinese text classification tasks. Furthermore, to select the most informative example for applying demonstration learning with the query sentence, we combine the cosine similarity and the mutual information and form a novel joint correlation scoring function. Experiment results conducted on five text classification tasks from CLUE illustrate the effectiveness against all discussed baselines. In addition, an extensive ablation study shows that the joint correlation scoring function is the most important component in the whole model. Though our proposal achieves notable improvements, finding suitable prompts for large-scale PLMs is not trivial, and carefully designed initialization of prompts is crucial. Our proposed template generation model requires the generation of a set of candidate templates that are used to cover the best possible performing templates. In addition, the pretrained model we use, BART, still introduces additional noise for template generation, which can degrade model performance. As for future work, we also would like to investigate the automation of choosing label words.

Author Contributions: Funding acquisition, D.L.; Project administration, T.S.; Validation, S.W. and H.C.; Writing—original draft, C.S.; Writing—review and editing, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Postgraduate Scientific Research Innovation Project of Hunan Province under No. CX20210068.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Appendix A. Confusion Metrics

Please note that we randomly select 1000 samples for each category in the test set on every corpus.

Table A1. Model performance. Confusion metric on AFQMCM.

AFQMC	Label 0	Label 1
Label 0	769	231
Label 1	225	775

Table A2. Model performance. Confusion metric on OCNLI.

OCNLI	Entailment	Neutral	Contradiction
entailment	782	159	59
neutral	120	608	272
contradiction	51	109	840

Table A3. Model performance. Confusion metric on INEWS.

INEWS	Label 0	Label 1	Label 2
Label 0	692	169	139
Label 1	221	764	15
Label 2	54	263	683

Table A4. Model performance. Confusion metric on BQ.

INEWS	Label 0	Label 1
Label 0	216	784
Label 1	754	246

Table A5. Model performance. Confusion metric on TNEWS.

TNEWS	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
T1	710	4	8	23	28	3	35	32	45	2	10	13	13	44	30
T2	24	769	14	1	10	12	1	5	9	3	101	24	5	4	18
T3	24	64	653	10	19	27	6	14	7	55	23	18	8	57	15
T4	1	40	2	739	3	29	25	12	38	20	9	38	18	24	2
T5	31	50	9	25	693	14	13	62	38	8	27	15	8	5	2
T6	5	41	2	16	21	745	5	21	67	14	34	4	8	7	10
T7	53	9	15	16	11	3	677	5	40	46	27	62	9	25	2
T8	10	13	18	13	37	54	41	682	26	2	5	48	24	22	5
T9	2	6	6	10	57	6	2	34	740	4	29	19	13	34	38
T10	32	18	6	4	31	17	21	37	5	741	8	2	34	18	26
T11	30	1	6	42	24	52	11	6	22	55	681	4	30	10	26
T12	1	22	12	86	16	41	9	24	3	70	6	623	44	1	42
T13	3	51	22	9	13	36	1	11	14	18	7	6	799	2	8
T14	15	1	32	9	5	4	12	91	66	36	11	18	83	582	35
T15	11	44	16	21	6	3	18	2	5	3	3	8	7	13	840

References

1. Lee, J.; Park, S. A Study on the Calibrated Confidence of Text Classification Using a Variational Bayes. *Appl. Sci.* **2022**, *12*, 9007. [CrossRef]
2. Ho, T.K.; Shih, W.Y.; Kao, W.Y.; Hsu, C.H.; Wu, C.Y. Analysis of the Development Trend of Sports Research in China and Taiwan Using Natural Language Processing. *Appl. Sci.* **2022**, *12*, 9006. [CrossRef]
3. Faralli, S.; Velardi, P. Special Issue on Social Network Analysis. *Appl. Sci.* **2022**, *12*, 8993. [CrossRef]
4. Zhang, H.; Wang, X.; Jiang, S.; Li, X. Multi-Granularity Semantic Collaborative Reasoning Network for Visual Dialog. *Appl. Sci.* **2022**, *12*, 8947. [CrossRef]
5. Saleh, H.; Mostafa, S.; Gabralla, L.A.; Aseeri, A.O.; El-Sappagh, S. Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models. *Appl. Sci.* **2022**, *12*, 8967. [CrossRef]
6. Vilares, D.; Alonso, M.A.; Gómez-Rodríguez, C. A linguistic approach for determining the topics of Spanish Twitter messages. *J. Inf. Sci.* **2015**, *41*, 127–145. [CrossRef]
7. Kim, Y.; Kim, J.H.; Lee, J.M.; Jang, M.J.; Yum, Y.J.; Kim, S.; Shin, U.; Kim, Y.M.; Joo, H.J. A pre-trained BERT for Korean medical natural language processing. *Sci. Rep.* **2022**, *12*, 1–10.
8. De Carvalho, V.D.H.; Costa, A.P.C.S. Towards corpora creation from social web in Brazilian Portuguese to support public security analyses and decisions. *Library Hi Tech* **2022**, ahead-of-print. [CrossRef]
9. Al-Maleh, M.; Desouki, S. Correction to: Arabic text summarization using deep learning approach. *J. Big Data* **2021**, *8*, 56. [CrossRef]
10. Mishra, A.; Shaikh, S.H.; Sanyal, R. Context based NLP framework of textual tagging for low resource language. *Multim. Tools Appl.* **2022**, *81*, 35655–35670. [CrossRef]
11. Zheng, J.; Cai, F.; Chen, W.; Lei, W.; Chen, H. Taxonomy-aware Learning for Few-Shot Event Detection. In Proceedings of the WWW '21—Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–3 April 2021; pp. 3546–3557.
12. Li, Z.; Ouyang, F.; Zhou, C.; He, Y.; Shen, L. Few-Shot Relation Classification Research Based on Prototypical Network and Causal Intervention. *IEEE Access* **2022**, *10*, 36995–37002. [CrossRef]
13. Qin, Y.; Zhang, W.; Zhao, C.; Wang, Z.; Zhu, X.; Shi, J.; Qi, G.; Lei, Z. Prior-knowledge and attention based meta-learning for few-shot learning. *Knowl. Based Syst.* **2021**, *213*, 106609. [CrossRef]
14. Perez, E.; Kiela, D.; Cho, K. True Few-Shot Learning with Language Models. *Adv. Neural Inf. Process. Syst. NIPS* **2021**, *34*, 11054–11070.
15. 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, EACL 2021, Online, 19–23 April 2021.
16. 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, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 2339–2352.
17. 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.
18. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021.

19. 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. *NIPS* **2020**, *33*, 1877–1901.
20. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
21. Dieng, A.B.; Wang, C.; Gao, J.; Paisley, J. TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. ICLR. 2017. Available online: https://openreview.net/pdf?id=rJbbOLcex (accessed on 30 October 2022).
22. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very Deep Convolutional Networks for Text Classification. 2017. Available online: https://aclanthology.org/E17-1104.pdf (accessed on 30 October 2022).
23. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *NIPS* **2014**, *27*, 3104–3112.
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
25. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. *NIPS* **2017**, *30*, 4077–4087.
26. Lyu, C.; Liu, W.; Wang, P. Few-Shot Text Classification with Edge-Labeling Graph Neural Network-Based Prototypical Network. COLING. ICCL. 2020. Available online: https://aclanthology.org/2020.coling-main.485.pdf (accessed on 30 October 2022).
27. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching Networks for One Shot Learning. *NIPS* **2016**, *29*, 3630–3638.
28. Yang, W.; Li, J.; Fukumoto, F.; Ye, Y. HSCNN: A Hybrid-Siamese Convolutional Neural Network for Extremely Imbalanced Multi-label Text Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, ACL, Punta Cana, Dominican Republic, 8–12 November 2020; pp. 6716–6722.
29. Wei, J.; Huang, C.; Vosoughi, S.; Cheng, Y.; Xu, S. Few-Shot Text Classification with Triplet Networks, Data Augmentation, and Curriculum Learning. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 5493–5500.
30. Howard, J.; Ruder, S. ACL. Universal Language Model Fine-Tuning for Text Classification. 2018. Available online: https://aclanthology.org/P18-1031.pdf (accessed on 30 October 2022).
31. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; pp. 1746–1751.
32. Abreu, J.; Fred, L.; Macêdo, D.; Zanchettin, C. *Hierarchical Attentional Hybrid Neural Networks for Document Classification*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 396–402.
33. Yang, L.; Zhang, M.; Li, C.; Bendersky, M.; Najork, M. Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching. In Proceedings of the CIKM '20: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 1725–1734.
34. Sun, S.; Sun, Q.; Zhou, K.; Lv, T. Hierarchical Attention Prototypical Networks for Few-Shot Text Classification. 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. 476–485.
35. Lai, V.D.; Nguyen, T.H.; Dernoncourt, F. Extensively Matching for Few-shot Learning Event Detection. In Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, NUSE@ACL 2020, Online, 9 July 2020; pp. 38–45.
36. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V.; et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
37. Devlin, J.; Chang, M.; 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; pp. 4171–4186.
38. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed.; Prentice Hall Series in Artificial Intelligence; Prentice Hall: Hoboken, NJ, USA, 2009.
39. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. 2020. Available online: https://aclanthology.org/2020.acl-main.703.pdf (accessed on 30 October 2022).
40. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 140:1–140:67.
41. Petroni, F.; Lewis, P.S.H.; Piktus, A.; Rocktaschel, T.; Wu, Y.; Miller, A.H.; Riedel, S. How Context Affects Language Models' Factual Predictions. In Proceedings of the Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, 22–24 June 2020.
42. Jiang, Z.; Anastasopoulos, A.; Araki, J.; Ding, H.; Neubig, G. X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5943–5959.
43. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. PTR: Prompt Tuning with Rules for Text Classification. *arXiv* **2021**, arXiv:2105.11259.
44. Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; Chen, H. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In Proceedings of the WWW '22: Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 2778–2788.
45. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT Understands, Too. *arXiv* **2021**, arXiv:2103.10385.
46. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. 2021. Available online: https://aclanthology.org/2021.acl-long.353.pdf (accessed on 30 October 2022).

47. Gu, Y.; Han, X.; Liu, Z.; Huang, M. PPT: Pre-trained Prompt Tuning for Few-shot Learning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 8410–8423.
48. Mishra, S.; Khashabi, D.; Baral, C.; Hajishirzi, H. Natural Instructions: Benchmarking Generalization to New Tasks from Natural Language Instructions. *arXiv* **2021**, arXiv:2104.08773.
49. Kumar, S.; Talukdar, P.P. Reordering Examples Helps during Priming-based Few-Shot Learning. 2021. Available online: https://aclanthology.org/2021.findings-acl.395.pdf (accessed on 30 October 2022).
50. Shao, Y.; Geng, Z.; Liu, Y.; Dai, J.; Yang, F.; Zhe, L.; Bao, H.; Qiu, X. CPT: A Pre-Trained Unbalanced Transformer for Both Chinese Language Understanding and Generation. *arXiv* **2021**, arXiv2109.05729.
51. Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; et al. CLUE: A Chinese Language Understanding Evaluation Benchmark. *arXiv* **2020**, arXiv:2004.05986.
52. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-Training With Whole Word Masking for Chinese BERT. *IEEE ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3504–3514. [CrossRef]
53. 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.