


## Article

# ConKgPrompt: Contrastive Sample Method Based on Knowledge-Guided Prompt Learning for Text Classification

Qian Wang <sup>1</sup> , Cheng Zeng <sup>2,3,\*</sup>, Bing Li <sup>4</sup> and Peng He <sup>3,5</sup>

<sup>1</sup> School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China; wqmaster@stu.hubu.edu.cn

<sup>2</sup> School of Artificial Intelligence, Hubei University, Wuhan 430062, China

<sup>3</sup> Key Laboratory of Intelligent Sensing System and Security, Hubei University, Wuhan 430062, China; penghe@hubu.edu.cn

<sup>4</sup> School of Computer Science, Wuhan University, Wuhan 430072, China; bingli@whu.edu.cn

<sup>5</sup> School of Cyber Science and Technology, Hubei University, Wuhan 430062, China

\* Correspondence: zc@hubu.edu.cn

**Abstract:** Text classification aims to classify text according to pre-defined categories. Despite the success of existing methods based on the fine-tuning paradigm, there is a significant gap between fine-tuning and pre-training. Currently, prompt learning methods can bring state of the art (SOTA) performance to pre-trained language models (PLMs) in text classification and transform a classification problem into a masked language modeling problem. The crucial step of prompt learning is to construct a map between original labels and the label extension words. However, most mapping construction methods consider only labels themselves; relying solely on a label is not sufficient to achieve accurate prediction of mask tokens, especially in classification tasks where semantic features and label words are highly interrelated. Therefore, the accurate prediction of mask tokens requires one to consider additional factors beyond just label words. To this end, we propose a contrastive sample method based on knowledge-guided prompt learning framework (ConKgPrompt) for text classification. Specifically, this framework utilizes external knowledge bases (KBs) to expand the label vocabulary of verbalizers at multiple granularities. In the contrastive sample module, we incorporate supervised contrastive learning to make representations more expressive. Our approach was validated on four benchmark datasets, and extensive experimental results and analysis demonstrated the effectiveness of each module of the ConKgPrompt method.

**Keywords:** prompt learning; contrastive learning; pre-trained language model



**Citation:** Wang, Q.; Zeng, C.; Li, B.; He, P. ConKgPrompt: Contrastive Sample Method Based on Knowledge-Guided Prompt Learning for Text Classification. *Electronics* **2023**, *12*, 3656. <https://doi.org/10.3390/electronics12173656>

Academic Editor: Fernando De la Prieta Pintado

Received: 3 August 2023

Revised: 23 August 2023

Accepted: 28 August 2023

Published: 30 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

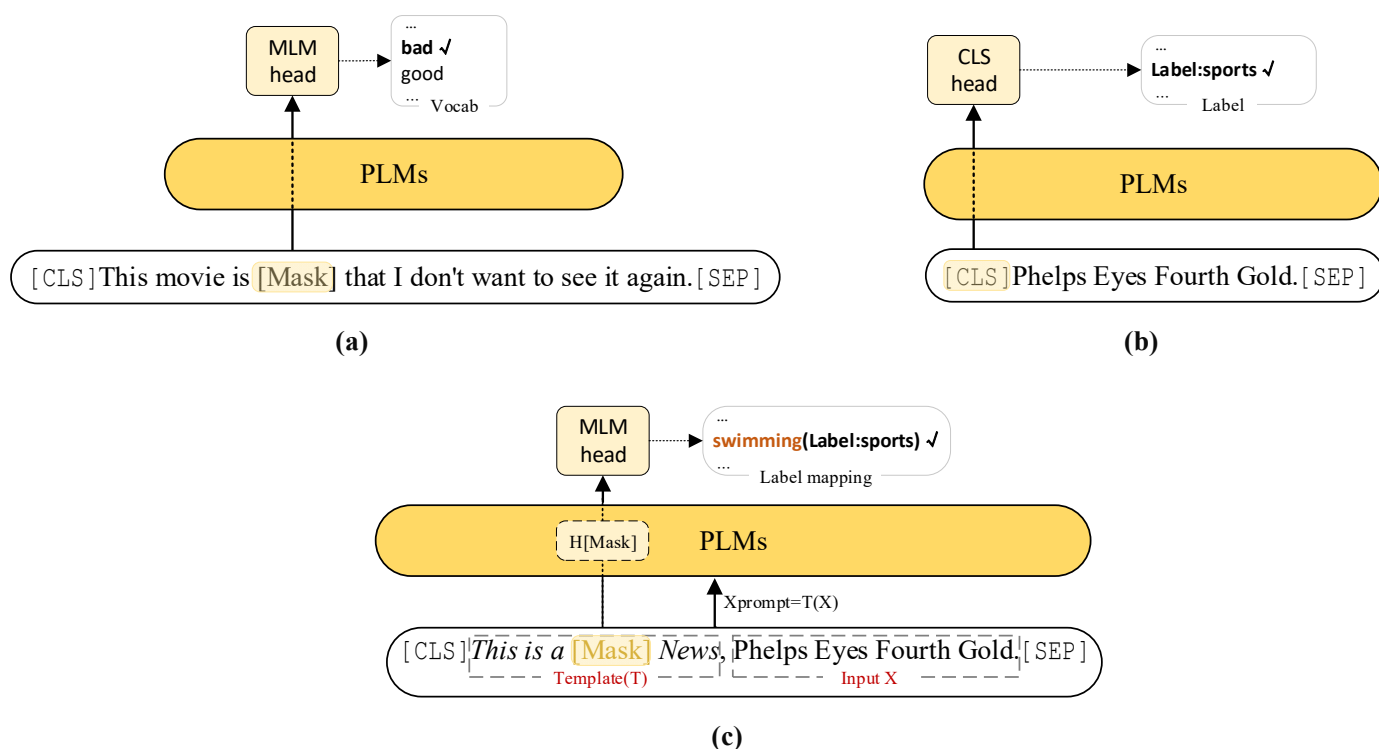
## 1. Introduction

In the era of the Internet, text is a crucial medium for the dissemination of information. The accurate classification of text based on semantic information is an important research direction, with significant implications for information utilization. Text classification is an important task in natural language processing. It is the foundation of many tasks [1,2] and has a wide range of applications in fields such as sentiment analysis [3], spam detection [4], and fake news detection [5]. Due to its expansive applicability, text classification demands substantial consideration and emphasis.

Traditional text classification methods utilize neural networks such as TextCNN [6] and TextRNN [7] to extract contextual features, leading to significant success in feature representation learning. Recently, pre-trained large language models such as BERT [8], RoBERTa [9], GPT [10], and ERNIE [11] have shown superior performance in NLP tasks. Researchers have attributed this to the prior knowledge accumulated during their pre-training. To take advantage of this prior knowledge and enhance classification performance, researchers have proposed a method of fine-tuning pre-trained language models (PLMs) for downstream tasks [12]. This method consists of adding a classification layer for feature

classification to the top layers of PLMs; using network structure and network parameters, the top layer is modified to adapt the downstream task output layer. Since PLMs have memorized vast amounts of raw corpus, they can offer useful knowledge for downstream tasks.

While the fine-tuning paradigm performs satisfactorily in downstream tasks that rely on large datasets, the increasing model size has raised the cost of fine-tuning a high-performing pre-trained language model (PLM) for downstream tasks. The performance bottleneck of fine-tuning has also hindered the ability of PLM-based text classification algorithms to be applied in real-world fields. To address this problem, prompt learning based on fine-tuning is proposed, inspired by GPT-3 [10]. This approach aims to promote consistency between the learning process and pre-trained objective via the usage of the corpus acquired during the pre-training process. As shown in Figure 1, compared with fine-tuning, prompt-tuning processes input text with a prompt template, converting it into a cloze-style phrase. During training, prompt-tuning only updates the template parameters to obtain supervised signals from downstream task training while keeping PLM parameters stable, which leads to higher prediction accuracy [13]. In prompt-tuning, mapping label words (such as “sport”) to specific categories (such as “swimming” or “football”) can efficiently decrease the disparities between the label and text spaces. This verbalizer construction strategy has been proven to be effective [14].



**Figure 1.** The general illustrations of pre-training, fine-tuning, and prompt-based learning. (a) The pre-training process uses mask strategies to pre-train PLMs; (b) the fine-tuning method uses a representation of the headers’ special token as a representation of the whole text for prediction; (c) the prompt-based fine-tuning (prompt-tuning) method allows the model to add a prompt template with a mask for prediction.

However, most existing work usually involves manually constructing the verbalizer, which is prone to errors during prediction. Some approaches combine optimization-based expansion words with a manual verbalizer [15], but in terms of label embedding, it only introduces words that are semantically close to the class name. KPT [16] is an efficient prompt-tuning method that combines external knowledge and various refinement techniques. However, this work on verbalizers has primarily been centered on the label

corresponding to the text, whereas text classification deals with the complexity and variety of the language and its extensive vocabulary containing semantically similar yet different phrases. Therefore, the development of a verbalizer cannot ignore the text background and conceptual information. Additionally, the versatility of prompt-tuning for varying tasks and datasets is limited. For instance, pre-trained language models may lack adequate awareness of specific subject areas. As a result, more labeled data are required to enhance the PLM's performance. Therefore, we attempt to use contrastive samples for knowledge-guided prompt-tuning.

We propose our ConKgPrompt model (contrastive sample method based on knowledge-guided prompt learning), which is an effective method of text classification. Initially, we encapsulate the original text via prompt engineering and develop a method for constructing and optimizing a cross-granularity verbalizer by introducing external knowledge bases targeting the implicit notions of the text and label words. Subsequently, we use the pre-trained language model ERNIE 3.0 [17] to extract semantic features. Furthermore, we introduce SupCon [18] into the current prompt learning process. Finally, we validate the effectiveness of each module of our method in the full data and few-shot scenarios. Our main contributions can be summarized as follow:

(1) In this work, we propose a knowledge-guided prompt-tuning method based on supervised contrastive learning, which can fully utilize the advantages of prompt learning to unleash the potential of ERNIE 3.0. To the best of our knowledge, this is the first work to adopt knowledge-guided prompt-tuning for text classification tasks.

(2) Introduced supervised contrastive learning can increase the effectiveness of the sample's learning during the training process. Moreover, integrated external knowledge bases can extract and integrate the knowledge information of sample and label words into an extended word set.

(3) We conducted extensive experiments on both real-world datasets and benchmark datasets. The experimental results demonstrate that our proposed model outperforms other methods in low-resource and full-data scenarios, highlighting the superiority and effectiveness of our method.

## 2. Related Works

### 2.1. Prompt Based Fine-Tuning

The essence of prompt learning is to unify downstream tasks as pre-training tasks to bridge the gap between objective forms in pre-training and fine-tuning. Pre-trained language models trained with prompt-tuning have proven to be effective for several NLP tasks, including text classification [19], relation classification [20], Twitter classification [21], and relation extraction [22].

As many studies have shown, the method of obtaining prompt templates greatly influences the performance of prompt learning. Liu et al. [13] proposed P-tuning, which targets optimized prompt token insertion on the input side to automatically identify knowledge templates in continuous space. Shin et al. [23] introduced gradient-guided searchING to automatically design prompts for diversified tasks. Gao et al. [14] selected the optimization of prompt token embedding, which builds on prompt-tuning to propose a small-sample fine-tuning method based on the language model. This method achieves good performance even in few-shot scenarios.

Verbalizer construction is another crucial component of prompt learning. PET [24] IS a conventional approach for prompt learning that demands less annotated data than traditional fine-tuning methods. However, PET's word expansion is unidirectional and fails to cover enough ground. Some studies have developed automatic verbalizers to search for better word expansions [25], but require large amounts of data. He et al. [16] proposed generating label expansion verbalizers using external knowledge bases (KBs). This work provides an essential basis for constructing prompt verbalizers in recent years. Nevertheless, for text classification tasks, the features contained in the text are highly similar to the labels and cannot be ignored.

## 2.2. Contrastive Learning

In numerous practical applications, obtaining a large volume of labeled data is challenging for researchers to achieve. To circumvent these challenges, researchers have introduced numerous techniques for performing deep feature extraction utilizing either unlabeled or weakly labeled samples. Contrastive learning has gained widespread attention due to its effectiveness in learning sequence data features. The core idea of contrastive learning is to compare the similarity of sample pairs in the feature space. By utilizing the labeled information of data samples, the distance between similar samples in the same feature space can be reduced during training, while the distance between dissimilar samples will be increased. Khosla et al. [18] introduced contrastive learning into the supervised scenario and achieved significant results in text classification tasks. SimCSE [26] implements sentence-level semantic representation of text based on contrastive learning and dropout. Unsupervised SimCSE constructs positive and negative samples with dropout, while supervised SimCSE constructs positive and negative samples using the contradiction labels contained in the NLI dataset. However, in the case of supervision, the scarcity of data does not aid in enhancing the semantic representation of sentences. Guo et al. [27] proposed a supervised word-weighted contrastive learning method that enhances source text data via the calculation of word weight and utilizes an adversarial approach in combination with supervised contrastive learning. However, this method requires a large amount of labeled data, and the methodology used in constructing positive and negative samples cannot guarantee semantic consistency.

## 2.3. Knowledge Utilization

In recent years, it has been demonstrated that utilizing external knowledge is a beneficial approach to enhancing PLM performance. These studies aim to utilize language models for knowledge representation learning that is typically used during the pre-training stage of PLMs and the fine-tuning stage of the model. However, the combination of external knowledge and pre-trained language models leads to heterogeneous embedding space and knowledge noise. ERNIE [11] introduced an encoder module for integrating knowledge into PLMs, which sets a precedent for knowledge graph fusion while preserving the traditional transformer structure. Liu et al. [28] aimed at the problems of PLMs and knowledge graph fusion. It is proposed that K-BERT infuses domain knowledge graphs into sentences, which is effective in Chinese NLP tasks.

Researchers are now exploring incorporating external knowledge into prompt learning. Liu et al. [29] proposed a prompt-tuning method enhanced with knowledge for identifying event causality in FSEC tasks. Background and relationship information gathered from external knowledge bases (KBs) were incorporated into the model. Additionally, attention mechanisms were designed to integrate the background and relationship information. Song et al. [30] proposed a trigger prompt-tuning method for few-shot event classification augmented with knowledge bases. Knowledge bases (KBs) are utilized to identify the triggers related to each granularity level of the event sentence, thereby resolving the issue of limited resources in classification tasks. These methods significantly enhance the performance of prompt-tuning. However, the approach is not easily transferable to other related tasks due to the high cost of model resources, the challenge of refining the candidate word construction, and neglect of the original text's information. Drawing inspiration from these methods, we integrate external knowledge bases into our ConKgPrompt method and develop a knowledge-guided prompt-tuning approach for classification tasks.

## 3. Preliminaries

This section provides some fundamental concepts on fine-tuning, prompt-tuning, and supervised contrastive learning before introducing our proposed method. We assume a text classification dataset,  $D = \{X, Y\}$ , where  $X$  represents a collection of input samples and  $Y$  are their respective labels. We will be utilizing this dataset as a foundation for the ensuing discussion.

### 3.1. Fine-Tuning Based on PLMs

For text classification tasks, the traditional fine-tuning method utilizes a language model  $M$  trained on a large corpus of text to initialize its own network parameters and then uses text data for training downstream tasks. Specifically, the input sample  $X$  is standardized into a sequence of  $[CLS]X[SEP]$ , where  $[CLS]$  is the word vector of the first word of the input text, which is used by the output layer to predict the category of the text, and  $[SEP]$  indicates the end of the sentence to serve as a boundary between two sentences. Then, using  $M$ , the sequence is encoded as hidden vectors  $E$ . By using an MLM head, we can obtain the probability distribution of the label set  $Y$ , as follows:

$$P = \text{softmax}(WE_{cls}) \quad (1)$$

### 3.2. Prompt-Tuning Based on PLMs

Prompt tuning is a method that introduces adaptable prompts to enhance the performance of a pre-trained language model  $M$  for specific tasks. The method begins by developing a prompt template  $T$  that includes  $[Mask]$  and an answer mapping vocabulary  $V$ .  $T$  is combined with an input text  $x \in X$  to create the final input sequence  $x_{prompt}$ , with the  $[Mask]$  token serving as a masked position for predicting the answer. Additionally, by creating a verbalizer function  $v(\cdot) : V \rightarrow Y$ , which can be defined as a map that links several individual words from the label word set  $V$  to the label space, the probability calculation for a specific label  $y$  can be obtained:

$$\begin{aligned} P(y | x) &= P([Mask] = v(y) | x) \\ &= \text{softmax}(W_{v(y)} \times E_{cls}) \\ &= \frac{\exp(W_{v(y)} \times E_{cls})}{\sum_{y' \in Y} \exp(W_{v(y')} \times E_{cls})} \end{aligned} \quad (2)$$

where  $W_v$  and  $E$  represent the hidden vectors of  $[Mask]$  and label Word  $V$ , respectively. We employ a binary sentiment classification task to better explain prompt-tuning. For example, we construct a template  $T(\cdot) = "This is [Mask]."$  We then define  $V$  as a transformation from the label space to several words in the vocabulary (e.g.,  $v(y = positive) \rightarrow good$ ,  $v(y = negative) \rightarrow bad$ ). Then, the input sequence is normalized to  $x_{prompt} = [CLS] x This is [Mask][SEP]$  using  $T$ , where the underscore sequence is a template previously defined. Finally, we utilize the probability of the predicted answer word (good or bad) to represent the corresponding probability of the class (positive or negative). Compared with traditional fine-tuning, prompt-tuning reuses the pre-training weights and does not introduce any new parameters.

### 3.3. Supervised Contrastive Learning

SupCon [18] is a specific method of contrastive learning. This method combines two augmented batches at the class level in the feature space. The loss function is designed based on the idea of adversarial training: shrinking the distance between anchors and positive samples in the feature space while increasing the distance between anchors and negative samples. The contrastive loss function for supervised tasks is designed as follows:

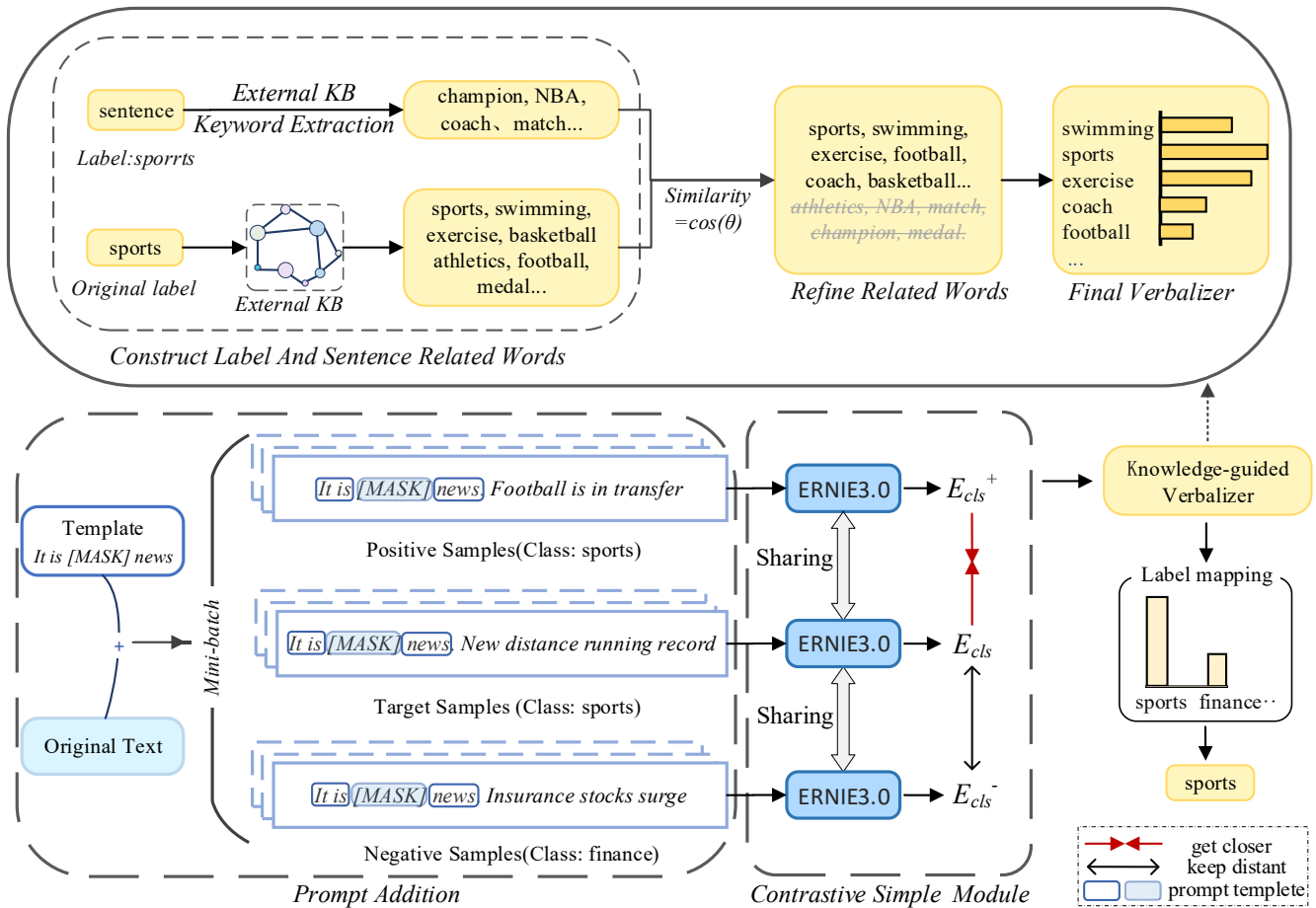
$$L_{sup} = \frac{1}{N} \sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P_i} -\log \frac{\exp(z_i \times z_p / \gamma)}{\sum_{a \in A_i} \exp(z_i \times z_a / \gamma)} \quad (3)$$

where  $\gamma$  is the temperature factor,  $P_i$  is the set of positive sample indexes,  $(\cdot)$  represents the cosine similarity on the hypersphere, and  $I$  contains all samples in a batch view.  $A$  is the set of indexes for the contrastive sample.

## 4. Related Theory

In this section, we begin by formulating the task of text classification, followed by a detailed description of our proposed ConKgPrompt model, as shown in Figure 2. This

approach comprises four modules, namely prompt engineering, supervised contrastive sample module, external knowledge acquisition, and ERNIE 3.0 for feature extraction.



**Figure 2.** Architecture of the ConKgPrompt.

#### 4.1. Prompt Engineering

Given that  $M$  is a language model pre-trained on a large-scale corpus, the dataset for the classification task is represented by  $D$ , with  $X$  representing the set of texts and  $Y$  representing the set of labels which correspond to each text. Each text corresponds to only one label. By integrating prompt-tuning, the classification task can instead be reframed as a masked language modeling task. Prompt engineering involves two main steps: prompt template construction and verbalizer creation.

##### 4.1.1. Prompt Template Construction

As indicated in Section 3.2, specific [MASK] tokens and templates can be employed to construct input instances for news classification tasks, and the corresponding prompt template can be designed as a cloze-style paradigm:

$$x_p = [\text{CLS}] \underline{\text{This is [MASK] News. They won the gold medal.}} [\text{SEP}] \quad (4)$$

The part that is underlined represents the template, and the [MASK] token represents the masked position that is predicted by the pre-trained language model. Using prompt-tuning, the input sequence is then encapsulated by the template.



#### 4.1.2. Verbalizer Construction

We construct a verbalizer as the mapping method to let the model predict the label with the highest relevance to [MASK]. The primary function of the verbalizer is to define the relationship between the label word set  $V$  and the label space  $Y$  in the vocabulary, represented by the function  $f(\cdot) : V \rightarrow Y$ . In the case of PLM  $M$ , the probability of filling the label word set into [MASK] can be expressed as:

$$P(y \in Y | x_p) = P([Mask] = v \in V_y | x_p) \quad (5)$$

By using the mapping relationship of  $f(\cdot)$ , the text classification task can be treated as a problem of label word probabilities. Taking inspiration from KPT [16], we consider the importance of expanding  $V$ , which is related to special categories with labels or words constructed automatically. To improve the performance of short text classification, in this study,  $\mathcal{V} = \{\text{sports}\}$  was expanded to  $\mathcal{V} = \{\text{athletics, sports, exercise, swimming, ...}\}$ . Considering the high semantic similarity and keyword prominence of text and labels, we will generate relevant words from both text and class names separately. By extracting keywords and retrieving the top- $N$  concepts that are relevant to the entities from an external knowledge base, and calculating the distance between selected concepts and class labels, the expanded label words can be further refined. The detailed steps for this process are presented in the next section.

#### 4.2. External Knowledge Acquisition

The central issue in text classification is to identify the words that have the greatest relevance to their respective labels across various aspects and granularities of the label space. For instance, when constructing the verbalizer in the prompt engineer, the “original label” is utilized as the [MASK] token to predict the masked token within a sentence. Nevertheless, this method lacks semantic coherence; thus, related words are generated from both the labels and text itself via unsupervised learning in the context of knowledge injection.

##### 4.2.1. Label Related Word Set Construction

We select Wantwords and CN-DBpedia as the external knowledge resources for our research. Wantwords is a high-performance knowledge base by THUNLP that is publicly available as an open-source project. Wantwords [31] and CN-DBpedia base [32] provide words that semantically match each specific instance’s description. They employ the label  $y$  to each category as an anchor word and select the top-ranked  $N$  concept words from the knowledge base, represented as  $N_{(Y)}$ . A set of words,  $V_y = \{y\} \cup N_{(Y)}$ , can be assigned to each label, mapping the label’s relevant word set to a conceptual class that includes or associates with it, utilizing a verbalizer.

##### 4.2.2. Text Keyword Extraction

We start the text keyword extraction process by preprocessing the text, which involves word segmentation and removal. Subsequently, we determine the frequency of each word appearing in the text and order the vocabulary by word frequency. The significance of a candidate keyword in the entire corpus can be determined using the unsupervised TF-IDF method. This method is a widely-used text representation approach that utilizes term frequency (TF) and inverse document frequency (IDF) scores to ascertain the significance of a particular word in the corpus. The computation of word frequency is illustrated below:

$$TF_i = \frac{n_i}{\sum_j n_j} \quad (6)$$

where  $\sum_j n_j$  is the number of occurrences of all words in the corpus, and  $n$  is the total number of occurrences of a given word  $t$ . Term frequency (TF) considers the significance of a word in the present text and a higher TF value indicates greater importance of the word

to the text. Inverse document frequency (IDF) represents the significance of a word to the entire corpus. It considers the frequency of a given word in the corpus. TF-IDF combines TF and IDF to calculate a comprehensive weight, which is computed using the following formula:

$$IDF_i = \log \frac{|X|}{1 + |\{j : t_i \in d_j\}|} \quad (7)$$

$$TFIDF_i = TF_i \times IDF_i \quad (8)$$

where  $|X|$  is the total number of sentences in the corpus. For each word, we calculate its TF-IDF score and sort it according to the score. A score greater than 0.5 is taken as the initial keyword candidate set.

To improve keyword quality and accuracy, we utilize CN-DBpedia as our primary knowledge base and utilize the example of keywords to extract concept words from the knowledge base with high correlation. Specifically, CN-DBpedia provides background information as a knowledge base and builds a graph where each graph node represents a natural language concept, and each edge corresponds to a semantic relation. We remove some edges as some semantic relations do not significantly impact keyword extraction.

#### 4.2.3. Verbalizer Refinement

Although we utilize external knowledge to refine the label words for multi-granularity expansion, we need to further refine the set to enhance quality label words that are impacted by conceptual deviations resulting from differences between the corpus used in the pre-training process and the knowledge bases.

The Word2vec [33] algorithm is designed for the acquisition of word vector representation. Moreover, it is also capable of assessing the correlation between words and word lists, measuring the degree of the relation of a word within a provided list. Initially, we load the word vector model  $\epsilon$  and implant the label within  $\epsilon$ . Subsequently, we utilize cosine similarity to calculate the similarity of each word  $y$  within the lexicon with the label word  $v$ :

$$\text{sim}(v, y) = \cos(\epsilon^v, \epsilon^y) \quad (9)$$

Additionally, we obtain the top-k extensions that have the closest semantic and part-of-speech proximity to  $v$  in embedding  $\epsilon$ .

#### 4.3. Knowledge Enhanced Pre-Training Model ERNIE 3.0

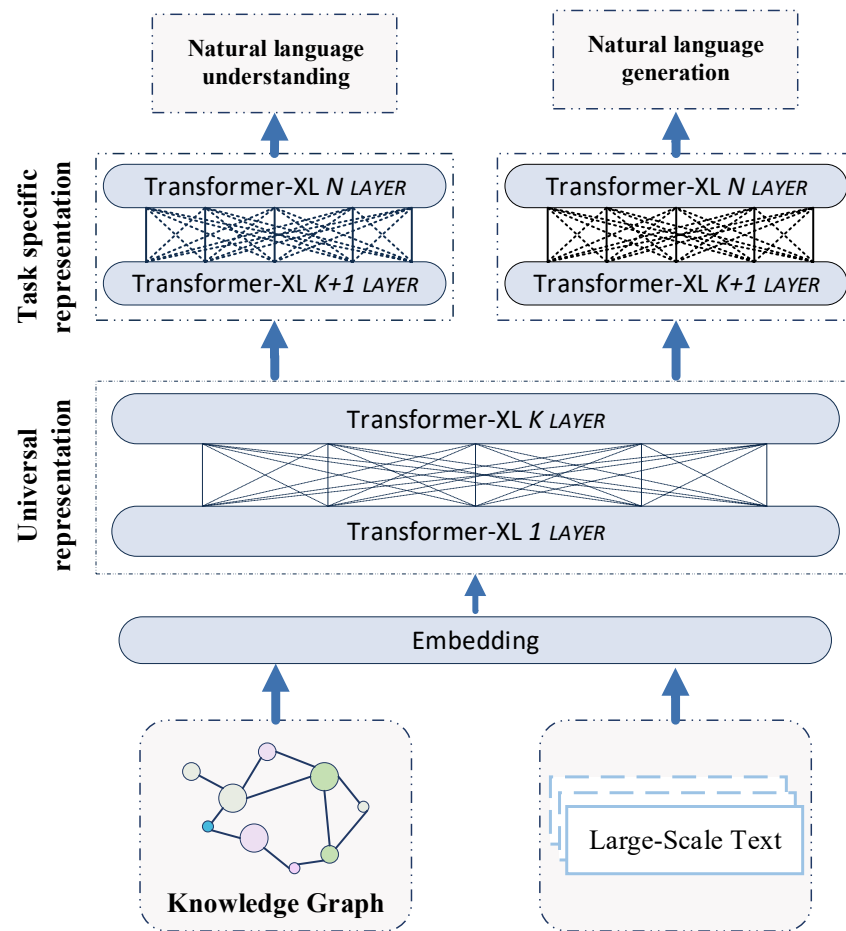
ERNIE 3.0 is a pre-trained language model for knowledge enhancement, which is trained using PaddlePaddle and contains billion-level parameters. It utilizes a fusion of auto-regressive networks and auto-encoding networks, trained on 4TB of pure text and large-scale knowledge graph corpuses. In the ERNIE 3.0 architecture, the universal representation module is employed to extract preliminary semantic features that can be used to understand and generate different tasks. Furthermore, task-specific representation modules are employed to extract task-specific semantic features learned based on the objectives of that particular task. With the combination of these powerful modules, ERNIE 3.0 demonstrates remarkable performance in sequence tasks. The structure of ERNIE 3.0 is shown in Figure 3.

ERNIE 3.0 employs multi-layer Transformer-XL [34] as the backbone of its structural framework. The input sequence  $x_p$  prompt-engineered, is encoded using the ERNIE 3.0 model as follows:

$$E = \text{ERNIE}(x_p) \quad (10)$$

where  $E \in R^{n \times d}$  is the feature representation of the input sequence,  $n$  is the number of tokens in the input sequence that include [CLS] and [SEP], and  $d$  is the hidden layer dimension of ERNIE 3.0.





**Figure 3.** The framework of ERNIE 3.0.

We used ERNIE 3.0 as our feature extractor for the following reasons. Firstly, ERNIE 3.0 enables the extraction of high-quality contextual representations, given its exceptional performance demonstrated in natural language processing tasks. Secondly, unlike other pre-trained language models (PLMs), ERNIE 3.0 incorporates external knowledge into the pre-training stage via the universal knowledge–text prediction task, which improves its memory and reasoning ability for pre-training corporuses. This characteristic aligns with our proposed knowledge-guided prompt-tuning method in this paper.

#### 4.4. Supervised Contrastive Simple Module

Many scholars have experimented with supervised contrastive learning methods to improve the effectiveness of sample features. In this paper, we utilize a supervised contrastive learning strategy, inspired by SupCon [18], to expand the program’s features by aligning explicit and implicit hidden representation with the same feature and emphasizing differences in the text embeddings with different label markings. The primary goal of supervised contrastive learning is to minimize the distance between similar embeddings and maximize it between dissimilar ones. For each  $(x, y)$  in batch  $\mathcal{B}$ , we extract representations  $E$  from the feature representation  $\varepsilon$  of  $x$ . Supervised contrastive learning loss is represented as follows:

$$P^{sup}(i) = \frac{\exp(\cos(E_i, E_p) / T)}{\sum_{a \in A_i^n, a \in i} \exp(\cos(E_i, E_a) / T)} \quad (11)$$

$$L_{sup} = \frac{1}{N} \sum_{i \in I} \frac{1}{|S_i^p(i)|} \sum_{p \in S_i^p} -\log P(i) \quad (12)$$

where  $P^{sup}(i, c)$  is the probability of similarity between  $E_i$  and  $E_p$ , and  $\mathcal{T}$  is an adjustable temperature parameter which is used to control the model's discrimination degree of negative samples.  $(cos)$  represents the cosine similarity on the hypersphere.  $\mathcal{A}_i^n$  is a set of negative samples not similar to  $i$ , and  $\mathcal{S}_i^p$  is a set of positive samples similar to  $i$ . To minimize Equation (11), we aim to maximize the numerator by learning the embedding of its positive sample  $i$ , and minimize the denominator by learning the embeddings of  $i$  and its negative samples.

Nevertheless, the original supervised contrastive learning is insufficient to tackle the overfitting issues of the classification task in this paper. To address this concern, we incorporate cross-entropy loss to alleviate the consequences of overfitting as shown in the following formula:

$$L_{CE} = -\frac{1}{N} \sum_{i \in I} \sum_{k \in K} y_{i,k} \log \frac{\exp(E_i \times W_k)}{\sum_{a \in A_i} \exp(E_i \times W_j)} \quad (13)$$

Finally, these two objectives can fully utilize the label information to achieve the goal of contrastive learning, and enhance the robustness and classification performance of ConKgPrompt. The total loss should be:

$$L = (1 - \lambda)L_{ce} + \lambda L_{sup} \quad (14)$$

where  $\lambda$  is a hyperparameter that controls the impact of the contrastive loss.

## 5. Experiments

To verify the effectiveness of our proposed model, we conducted several experiments on both few-shot and full-data text classification tasks. This section provides detailed descriptions of our dataset construction process, experimental settings, baseline methods, main results, and ablation study.

### 5.1. Dataset and Templates

We conduct experiments using the following datasets. Specifically, we select four datasets with different tasks and sentence lengths to demonstrate the effectiveness of our approach, including topic classification, and question classification. The specific information for each dataset is provided in Table 1. Below, we introduce the sources and original sizes of each dataset, as well as our methods for processing the original datasets.

**Table 1.** Dataset statistics.

Dataset	Task Type	Classes	Average Length	Training	Test
THUCNews	Topic classification	6	54	48,000	6000
Tnews	Topic classification	15	65	1185	2010
Healthcare	Question classification	5	46	10,800	1906
Onshop	Sentiment classification	2	78	50,220	6275

**THUCNews:** The THUCNews (<http://thuctc.thunlp.org/THUCNews>, accessed on 25 January 2016) dataset is sourced from historical data from Sina news subscription channels spanning the period from 2005 to 2011. Data cleaning is conducted on the massive dataset, resulting in 60,000 records which are integrated and categorized into 6 candidate categories: finance, stock, science, society, politics, and entertainment. Each category consists of 10,000 records.

**Tnews:** The Tnews Chinese text classification dataset is sourced from FewCLUE [35], a benchmark for evaluating natural language understanding in Chinese. It contains 15 categories, including military, sports, and education, and is derived from the news section

of the Toutiao app. The single training set comprises 240 samples, while the complete training set includes 1185 samples.

**Healthcare:** We utilize a crawler tool to extract question texts from mainstream medical websites in China and manually annotate the resulting data. Based on this extensive collection of data, we construct the healthcare dataset, comprising 12,706 items categorized into five distinct categories: disease diet, seasonal diet, sports and fitness, weight loss and beauty, and dietary contraindications.

**Onshop:** The online shopping evaluation dataset ([https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/online\\_shopping\\_10\\_cats](https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/online_shopping_10_cats), accessed on 24 March 2018) comes from the review data of various e-commerce platforms and collates the review data from ten commodity categories: books, tablets, mobile phones, fruits, shampoo, water heaters, milk, clothes, computers, and hotels, totaling 56,495 pieces of data, with about 28,000 positive and negative reviews each.

In order to reduce the influence of template on experimental results, we design four templates for each data set, as shown in Table 2.

**Table 2.** The templates of each dataset.

Datasets	Template
THUCNews	A [MASK] news: <Input> It is a [MASK] news: <Input> <Input> This topic is about [MASK] [News: [MASK]] <Input>
Tnews	A [MASK] news: <Input> It is a [MASK] news: <Input> <Input> This topic is about [MASK] [News: [MASK]] <Input>
Healthcare	A [MASK] question: <Input> It is a [MASK] question: <Input> <Input> This topic is about [MASK] [Category: [MASK]] <Input>
Onshop	It was [MASK]: <Input> This was [MASK]: <Input> Just [MASK]: <Input> <Input> All in all, it was [MASK] <Input> In summary, it was [MASK]

## 5.2. Baselines

**Fine-tuning.** Traditional fine-tuning methods involve obtaining the hidden embedding of the [CLS] token via a pre-trained language model (PLM) and then using a classification layer for prediction.

**DualCL.** Chen et al. [36] propose a unique data augmentation method by adding corresponding labels to the input sentence, obtaining the label token embedding and [CLS] token embedding through pre-trained language models.

**PET.** Schick et al. [24] propose the use of manual templates to form prompts, add a prefix or suffix to the input text and mask out some token, and convert the input instances into fill-in-the-blank phrases to help the language model understand the previously given task.

**P-tuning.** Liu et al. [13] propose the learning of continuous prompts by inserting trainable variables for improving the performance of the model on specific tasks.

**Soft Verbalizer (Soft).** The Soft verbalizer is proposed by WARP [37], which uses continuous prompts for each class and uses an MLM head to output the dot product between the class vector and the input as the probability of each class.

**Knowledgeable Prompt-tuning (KPT).** Hu et al. [16] extend the verbalizer using external knowledge bases such as related word lists and sentiment dictionaries to expand the initially defined label words based on manual definitions.

### 5.3. Experiment Settings

We conducted all our experiments using PyTorch on an NVIDIA GeForce RTX5000 environment. We use CN-DBpedia and Wantwords as the external knowledge base to obtain labels' relevant words. For the full data scenario, we experimented with different prompt methods based on ERNIE 3.0-Xbase. In the few-shot scenario, we use ERNIE 3.0 as the pre-trained language model to compare with other advanced prompt-based methods. We randomly extract  $k = 1, 5, 10$ , or 20 instances from the original training set to form the training set using four equally sized sub-sets. We use accuracy as the evaluation metric for all experiments and use the best model for testing. To reduce the randomness of experimental results, we repeat the experiments five times to obtain the average score and standard deviation of the test results and provided the optimal results after each experiment. We conducted all experiments for 20 epochs and select the most effective verification points to test.

### 5.4. Main Results

We evaluate and analyze the proposed ConKgPrompt method in the full-data scenario and few-shot scenarios and compare our method with the fine-tuning-based method and the prompt-tuning-based method. The experimental results are shown in Tables 2 and 3. “w/o SCL” means not using supervised contrast learning, while “w/SCL” refers to our proposed method that uses supervised contrastive learning. This setting can be regarded as an ablation experiment to analyze the influence of supervised contrastive learning on the experiment. To minimize the effects of randomness in the results, each experiment was repeated five times, and the final results are reported as the average score and standard deviation. Additionally, the best performance achieved in each experiment is indicated in parentheses.

**Table 3.** Accuracy scores (%) on three datasets in the full-data scenario.

Methods		THUCNews	Healthcare	Onshop
ERNIE 3.0-Xbase	Fine-tuning	93.22 $\pm$ 0.34 (93.4)	92.69 $\pm$ 0.55 (93.0)	94.46 $\pm$ 0.65 (95.1)
	DualCL	93.41 $\pm$ 0.23 (93.5)	92.92 $\pm$ 0.36 (93.2)	94.62 $\pm$ 0.78 (95.3)
	PET	93.27 $\pm$ 0.41 (93.6)	92.63 $\pm$ 0.31 (93.1)	94.68 $\pm$ 0.83 (95.4)
	P-tuning	93.53 $\pm$ 0.24 (93.8)	92.47 $\pm$ 0.41 (92.8)	94.87 $\pm$ 0.71 (95.5)
	Soft	93.32 $\pm$ 0.35 (93.6)	92.68 $\pm$ 0.51 (93.3)	94.92 $\pm$ 0.60 (95.5)
	Ours w/o SCL	93.83 $\pm$ 0.30 (94.1)	93.10 $\pm$ 0.68 (94.3)	94.95 $\pm$ 0.61 (95.6)
	Ours	<b>94.04 <math>\pm</math> 0.16 (94.5)</b>	<b>93.84 <math>\pm</math> 0.64 (94.8)</b>	<b>95.67 <math>\pm</math> 0.54 (96.2)</b>

The ConKgPrompt data are bolded for comparison.

#### 5.4.1. Full-Data

From Table 3, in the full-data scenario, most of the methods based on prompt-tuning are better than those based on fine-tuning, which indicates that prompt learning can better release the potential of pre-trained models. The method proposed in our paper consistently outperforms the fine-tuning method and the prompt-based method. Compared to the ERNIE 3.0-based most advanced method, our ConKgPrompt method achieved relative improvements of 0.51%, 0.92%, and 0.8% in three different domain datasets, respectively, due to its effective utilization of external knowledge and pre-training corpus.

#### 5.4.2. Few-Shot

From Table 4, the results of the prompt-based methods improve when the experimental shots are set from 1-shot to 20-shot, and the gap between the baseline methods gradually decreases, especially in the 20-shot experiments, the results of the baseline methods do not differ much. Nevertheless, our method still outperforms the baseline approach. It is shown that knowledge-guided prompt learning appropriately utilizes the prior knowledge stored in PLMs to adapt to the task while effectively exploiting the sample variability via

supervised contrastive learning. In THUCNews, Onshop, and Tnews with better annotation quality, KPT still outperforms other baseline methods, but the results are unstable, which is because KPT can effectively expand label words but has limited ability to release PLM pre-training knowledge and inadequately covers text. Our ConKgPrompt expands label words with multiple granularities, and in terms of variance, ConKgPrompt has smaller error than the baseline method in most cases, and the variance decreases as shot increases, indicating that we consider the text when constructing label words to make the training process stable.

**Table 4.** Accuracy scores (%) on three datasets in few-shot scenario.

k-Shot	Methods	THUCNews	Tnews	Healthcare	Onshop
1-shot	PET	51.75 ± 10.20 (60.5)	39.40 ± 3.73 (43.1)	53.73 ± 6.61 (60.8)	52.11 ± 10.32 (59.9)
	P-tuning	47.78 ± 5.34 (52.9)	38.96 ± 5.16 (44.9)	61.76 ± 5.57 (70.1)	48.71 ± 5.56 (54.9)
	Soft	48.91 ± 1.97 (50.5)	40.88 ± 6.17 (46.2)	61.26 ± 9.59 (71.9)	48.14 ± 4.98 (55.2)
	KPT	57.43 ± 3.03 (62.3)	45.11 ± 4.81 (45.8)	60.59 ± 7.20 (67.2)	58.23 ± 3.54 (62.1)
	Ours w/o SCL	58.58 ± 3.73 (61.1)	45.97 ± 1.72 (48.1)	63.61 ± 8.81 (73.8)	59.31 ± 7.11 (66.8)
	Ours	<b>61.19 ± 3.79 (67.5)</b>	<b>46.80 ± 1.87 (49.1)</b>	<b>64.01 ± 9.41 (75.8)</b>	<b>62.31 ± 8.42 (71.1)</b>
5-shot	PET	74.96 ± 2.88 (79.1)	49.95 ± 2.71 (52.4)	72.44 ± 2.19 (73.8)	73.87 ± 2.75 (78.1)
	P-tuning	71.02 ± 3.25 (74.9)	49.87 ± 2.20 (52.1)	79.92 ± 2.96 (83.3)	74.45 ± 3.53 (79.2)
	Soft	77.95 ± 2.25 (79.6)	52.72 ± 0.63 (53.5)	79.11 ± 4.98 (82.7)	73.54 ± 2.94 (78.6)
	KPT	77.45 ± 2.37 (80.1)	52.54 ± 1.77 (53.9)	81.39 ± 3.19 (83.9)	75.86 ± 2.43 (79.4)
	Ours w/o SCL	79.67 ± 2.29 (81.7)	53.10 ± 1.21 (54.1)	81.95 ± 2.08 (84.3)	78.13 ± 2.78 (81.4)
	Ours	<b>81.45 ± 1.75 (83.5)</b>	<b>53.62 ± 1.06 (54.7)</b>	<b>82.15 ± 2.83 (84.9)</b>	<b>81.94 ± 2.36 (84.3)</b>
10-shot	PET	79.82 ± 2.34 (82.8)	52.92 ± 1.71 (55.3)	81.32 ± 1.61 (83.4)	79.11 ± 2.97 (82.2)
	P-tuning	80.96 ± 2.14 (83.2)	52.76 ± 1.83 (55.1)	83.17 ± 1.59 (84.8)	80.42 ± 2.67 (83.8)
	Soft	82.82 ± 1.31 (83.9)	53.19 ± 1.40 (55.2)	83.01 ± 2.14 (84.1)	80.11 ± 2.32 (82.6)
	KPT	81.91 ± 2.27 (83.6)	53.74 ± 1.74 (55.4)	82.39 ± 1.60 (84.6)	81.98 ± 2.24 (84.1)
	Ours w/o SCL	82.65 ± 1.55 (83.5)	54.29 ± 1.07 (55.8)	83.51 ± 1.31 (84.9)	82.87 ± 1.57 (84.5)
	Ours	<b>84.61 ± 0.29 (84.9)</b>	<b>54.43 ± 1.22 (55.9)</b>	<b>83.98 ± 1.01 (85.2)</b>	<b>84.18 ± 1.81 (85.1)</b>
20-shot	PET	84.84 ± 1.41 (85.9)	54.60 ± 1.15 (55.3)	84.61 ± 0.92 (85.3)	84.34 ± 1.52 (85.9)
	P-tuning	85.16 ± 1.65 (86.4)	53.81 ± 1.39 (55.7)	86.08 ± 0.79 (86.9)	85.36 ± 1.15 (86.5)
	Soft	84.59 ± 0.61 (85.3)	54.64 ± 0.73 (55.4)	85.62 ± 0.62 (86.3)	84.79 ± 1.35 (86.1)
	KPT	85.03 ± 0.97 (85.8)	55.34 ± 0.67 (55.8)	86.13 ± 0.58 (86.8)	86.36 ± 0.65 (86.9)
	Ours w/o SCL	86.53 ± 0.21 (86.8)	55.85 ± 0.38 (55.9)	86.54 ± 0.62 (87.4)	86.97 ± 0.95 (87.8)
	Ours	<b>87.42 ± 0.24 (87.7)</b>	<b>56.12 ± 0.57 (56.4)</b>	<b>87.25 ± 0.72 (88.4)</b>	<b>87.32 ± 0.82 (88.2)</b>

The ConKgPrompt data are bolded for comparison.

### 5.5. Ablation Study

We conduct ablation experiments to verify the validity of our proposed components. For the ablation study, w/o SCL, w/o EKA, and w/o BOTH are the variants of ConKgPrompt that does not conduct SCL (supervised contrastive learning), EKA (external knowledge acquisition) and both SCL and EKA, respectively. The effectiveness of the added modules is proven by the increase in our method's performance when each module is incorporated, as shown in Tables 3 and 5 in 10-shot and 20-shot settings; the worst overall performer is w/o BOTH. The experimental results show that both SCL and EKA positively affect ConKgPrompt. Moreover, we observe that the performance improvement is more significant in few-shot scenarios and that our module improved ConKgPrompt more significantly with more samples compared to different shot settings. Of course, the improvement in the method also depends on the efficiency of the PLM release, which shows that the module we added gives PLM a considerable performance improvement in low resource settings.

**Table 5.** Ablation study of ConKgPrompt in 10/20 few-shot scenarios.

k-Shot	Methods	THUCNews	Tnews	Healthcare	Onshop
10-shot	ConKgPrompt	<b>84.61 ± 0.29 (84.9)</b>	<b>54.43 ± 1.22 (55.9)</b>	<b>83.98 ± 1.01 (85.2)</b>	<b>84.18 ± 1.81 (85.1)</b>
	w/o SCL	82.65 ± 1.55 (83.5)	54.29 ± 1.07 (55.8)	83.51 ± 1.31 (84.9)	82.87 ± 1.57 (84.5)
	w/o EKA	82.42 ± 1.23 (83.6)	54.14 ± 1.34 (55.4)	82.89 ± 1.54 (84.8)	82.57 ± 1.04 (83.7)
	w/o BOTH	81.36 ± 1.84 (83.1)	53.46 ± 1.61 (55.1)	82.57 ± 1.61 (84.1)	81.89 ± 1.68 (83.6)
20-shot	ConKgPrompt	<b>87.42 ± 0.24 (87.7)</b>	<b>56.12 ± 0.57 (56.4)</b>	<b>87.25 ± 0.72 (88.4)</b>	<b>87.32 ± 0.82 (88.2)</b>
	w/o SCL	86.53 ± 0.21 (86.8)	55.85 ± 0.38 (55.9)	86.54 ± 0.62 (87.4)	86.97 ± 0.95 (87.8)
	w/o EKA	86.01 ± 0.34 (86.2)	55.94 ± 0.52 (56.2)	86.32 ± 0.77 (87.2)	86.51 ± 1.35 (87.8)
	w/o BOTH	85.83 ± 0.81 (86.7)	55.63 ± 0.58 (55.9)	86.37 ± 0.61 (86.9)	85.98 ± 1.52 (87.4)

The ConKgPrompt data are bolded for comparison.

## 6. Analysis

In this section, we will analyze in more detail why supervised contrastive learning and knowledge-guided prompt learning are useful for text classification tasks.

### 6.1. Performance Influence between Different Pre-Trained Language Models

In this subsection, we explore the influence of different pre-trained language models on experimental results. We conducted experiments on BERT-base, RoBERTa-base, ERNIE 3.0-base, and ERNIE 3.0-Xbase. The experiment scenario is full-data so it can fully reflect the performance of PLMs. The experimental results are shown in Table 6. The averaged values of the four models on the three datasets are 93.35%, 93.49%, 93.90%, and 94.52%, respectively. We find that RoBERTa-base is 0.14% higher than BERT-base. ERNIE 3.0-Xbase outperforms RoBERTa-base by 1.03%. ERNIE 3.0 learned more than RoBERTa and BERT did in the pre-training phase. Therefore, using ERNIE 3.0-Xbase for knowledge-guided prompt tuning is more effective. In addition, we find that ERNIE 3.0-Xbase is 0.62% higher than ERNIE 3.0-base. This phenomenon illustrates that for our ConKgPrompt method, the larger the PLM size, the better the method will work on the downstream task.

**Table 6.** Result of ConKgPrompt on different pre-trained language model in full-data scenarios.

PLM		THUCNews	Healthcare	Onshop	Avg
BERT	base	92.78 ± 0.16 (92.9)	92.49 ± 0.23 (92.5)	94.78 ± 0.24 (95.1)	93.35
RoBERTa	base	93.19 ± 0.24 (93.3)	92.41 ± 0.21 (92.6)	94.87 ± 0.31 (95.2)	93.49
ERNIE 3.0	base	93.41 ± 0.19 (93.5)	92.77 ± 0.27 (93.0)	95.51 ± 0.78 (96.3)	93.90
	Xbase	<b>94.04 ± 0.16 (94.5)</b>	<b>93.84 ± 0.64 (94.8)</b>	<b>95.67 ± 0.54 (96.2)</b>	<b>94.52</b>

The ConKgPrompt data are bolded for comparison.

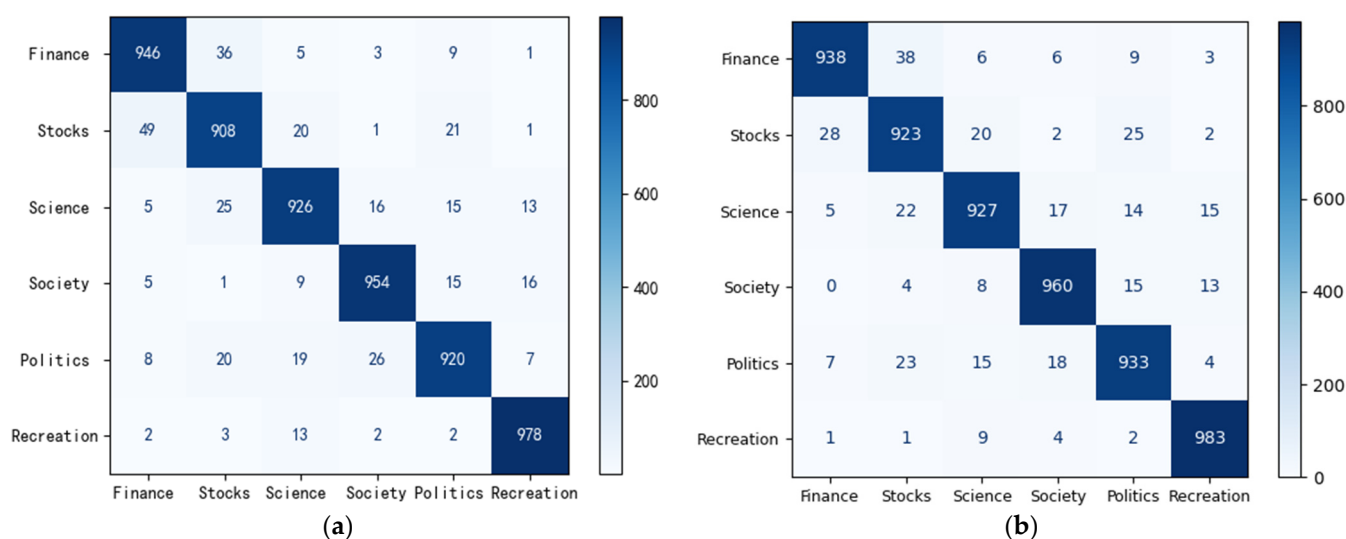
### 6.2. Impact of Knowledge-Guided Prompt Learning

A confusion matrix is a tool used to evaluate the performance of a model, and its primary metrics are presented in Table 7. It can clearly show the correspondence between the model's predictions and the ground truth labels across different categories. To explore the impact of incorporating knowledge on the prediction of each class in the dataset, we plot the test results of P-tuning and ConKgPrompt (w/SCL) under the full-data scenario, as shown in Figure 4.

**Table 7.** The first order index of the confusion matrix.

	Actual Positive	Actual Negative
Predict Positive	True Positive (TP)	False Positive (FP)
Predict Negative	False Negative (FN)	True Negative (TN)





**Figure 4.** Comparison of confusion matrices between P-tuning and ConKgPrompt in the full-data scenario of THUCNews dataset: (a) P-tuning based on ERNIE 3.0; (b) ConKgPrompt.

We use ERNIE 3.0-Xbase as the feature extractor in order to remove the influence of PLMs on the introduction of knowledge prompt. From Figure 4a, we can see that the prompt-tuning-based approach can achieve good classification results, but it is difficult to distinguish between the Finance and Stocks categories due to some similarities in the semantic features of these two categories. We find from Figure 4b by using knowledge-guided prompt-tuning that each category has more accurate predictions than P-tuning, except for the “finance” category, and the overall number of correct predictions increases for both finance and stocks. We deduce that this is from the bias of the categorization weights due to the crossover of extended words’ meanings in the prediction of labeled words extended by external knowledge in both categories, so our future work will also focus on confusable text extension refinement and exploring PLM for word perception and understanding. This shows we introduced external knowledge of prompt-tuning to improve the classification effect overall. This is in line with our analysis above.

### 6.3. Impact of Contrastive Sample Module

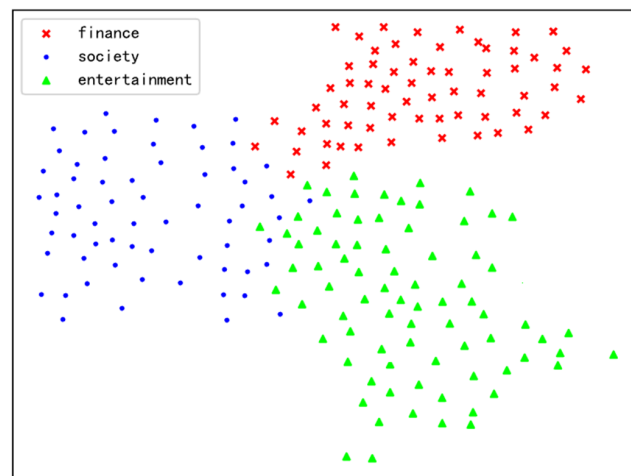
To further explore the ability of contrastive learning to improve the quality of representation, we select partial data from the THUCNews test dataset in the categories of finance, society, and entertainment, and use t-SNE (t-distributed stochastic neighbor embedding) to create a visual representation of the feature distribution under the 20-shot scenario, as shown in Figure 5.

From Figure 5, we clearly find that the distance between the same classes is reduced, demonstrating that the joint contrast loss uses the relationship between training samples to aggregate similar samples in the feature space, which can make the classifier more discriminative.

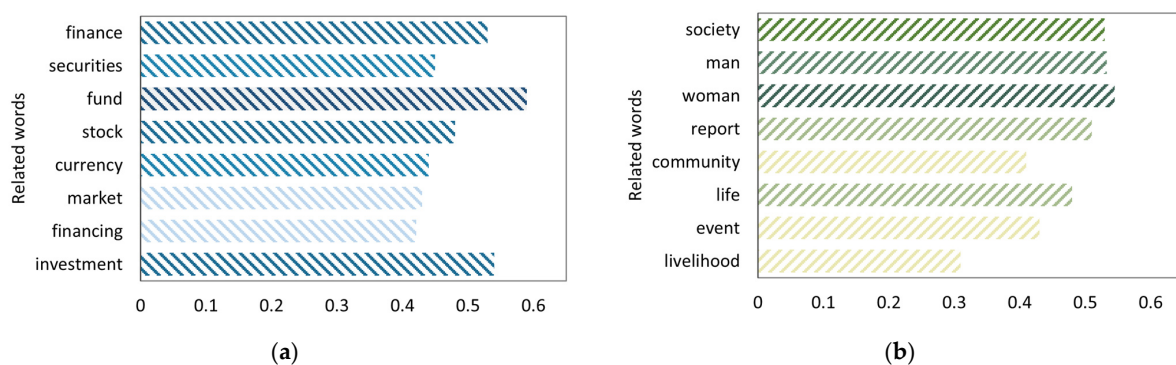
### 6.4. Visualization of Label Related Word Weights

In order to understand the role of EKA more intuitively, we also analyze the weight distribution of each related word after training. In the experiment, we average weights in a few-shot setting. As shown in Figure 6, we show some news tag words filtered via EKA, among which finance and society have higher weight, which is consistent with our intuition. As shown in Figure 6a, “fund” has the highest weight. If the relevant words are selected only by intuition, “fund” may be omitted. Moreover, “financing” should have a higher weight intuitively, but in fact has the lowest weight. As shown in Figure 6b, “woman” and “man” have a high weight, which indicates that these keywords are likely to be filtered out if they are simply extended from the external knowledge graph without considering the

sentence itself. These results suggest that PLMs may differ from how humans perceive and define words, and that PLMs understand words more in terms of semantics and relevance.



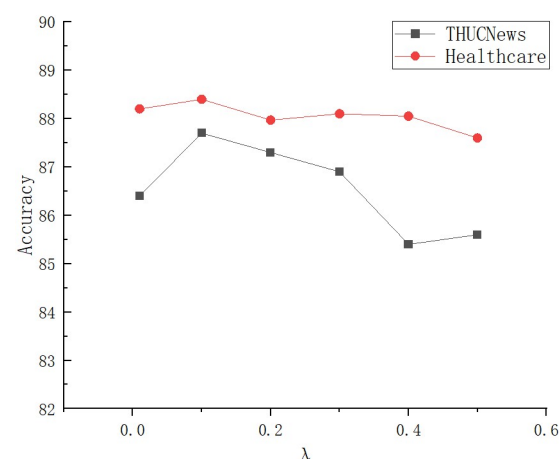
**Figure 5.** t-SNE visualization of vectors over the low-resource of THUCNews test set.



**Figure 6.** The weight distribution of label-related words in THUCNews: (a) the weights of finance; (b) the weights of society.

### 6.5. Hyperparameter Influence

In order to explore the influence of the joint contrastive learning parameter  $r$  on the experimental results, we set  $\lambda \in \{0.01 : 0.5\}$  on THUCNews to analyze the influence on the classification effect of ConKgPrompt. As shown in Figure 7, our model obtains the best result when the scaling coefficient is set to 0.1.



**Figure 7.** Fluence of SupCon and CE fusion ratio.

## 7. Discussion

We proposed ConKgPrompt for text classification as efficient in both few-shot and full-data scenarios. Our method can be dissected into the following processes: first, amalgamating the input text with the prompt template to create a new sequence; next, incorporating an external knowledge base to construct label- and sentence-related words, ultimately confirming the definitive verbalizer; and lastly, acquiring features from the supervised contrastive module by learning from the output of the PLMs. As demonstrated in the main results, it becomes apparent that our proposed model exhibits minimal variance when applied to datasets characterized by robust labeling quality. Conversely, datasets featuring limited data or intricate semantics showcase higher variance. This disparity underscores the susceptibility of ConKgPrompt to diverse dataset attributes, implying that there is indeed scope for further enhancement. Addressing this facet will remain an integral component of our future endeavors. In this vein, we are keen on capitalizing on the prowess of unsupervised learning to capture more universally applicable features. By imbuing the training process with unsupervised objectives, we anticipate augmenting the model's capacity to accommodate varied scenarios and contexts, thereby achieving a higher degree of generalization.

## 8. Conclusions

In this study, we proposed a contrastive sample method based on knowledge-guided prompt learning framework (ConKgPrompt) for text classification. ConKgPrompt utilizes external knowledge bases for multi-granularity construction of verbalizers and refines the verbalizer based on Word2vec and similarity. We employed ERNIE 3.0, which has rich prior knowledge, for feature extraction, and the training strategy utilizes supervised contrastive learning to exploit prior differences in samples to enhance sample representation. Experimental results and visual analyses indicate that the ConKgPrompt method achieves better classification performance than previous studies, particularly in the few-shot settings, and outperforms existing prompt-tuning methods, demonstrating the robustness and effectiveness of ConKgPrompt. Future work will focus on studying prompt continuous template generation strategies that can effectively express the grammatical relationships of text, and consider the use of unsupervised contrastive learning to enhance model representation.

**Author Contributions:** Conceptualization, Q.W.; methodology, Q.W.; software, Q.W.; formal analysis, Q.W.; visualization, Q.W.; writing-original draft preparation, Q.W.; writing-review and editing, C.Z.; resources, C.Z. and B.L.; funding acquisition, C.Z. and P.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the Key R & D projects in Hubei Province (No. 2021BAA188, 2021BAA184, 2022BAA044), the National Natural Science Foundation of China (No. 62102136), the Science and Technology Innovation Program of Hubei Province (No. 2020AEA008).

**Data Availability Statement:** The data presented in this study can be provided upon request.

**Acknowledgments:** The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation.

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

## References

1. Tang, J.W.; Li, R.X.; Wang, K.P.; Gu, X.W.; Xu, Z.Y. A Novel Hybrid Method to Analyze Security Vulnerabilities in Android Applications. *Tsinghua Sci. Technol.* **2020**, *25*, 589–603. [\[CrossRef\]](#)
2. Zhao, B.; Zhao, P.Y.; Fan, P.R. ePUF: A Lightweight Double Identity Verification in IoT. *Tsinghua Sci. Technol.* **2020**, *25*, 625–635. [\[CrossRef\]](#)
3. Pagolu, V.S.; Reddy, K.N.; Panda, G.; Majhi, B. Sentiment analysis of Twitter data for predicting stock market movements. In Proceedings of the 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), Paralakhemundi, India, 3–5 October 2016; pp. 1345–1350.
4. Debnath, K.; Kar, N. Email spam detection using deep learning approach. In Proceedings of the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 26–27 May 2022; pp. 37–41.

5. Guo, Y.; Lamaazi, H.; Mizouni, R. Smart edge-based fake news detection using pre-trained BERT model. In Proceedings of the 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 10–12 October 2022; pp. 437–442.
6. Kim, Y.J. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
7. Liu, P.; Qiu, X.; Huang, X.J. Recurrent neural network for text classification with multi-task learning. *arXiv* **2016**, arXiv:1605.05101.
8. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.J. 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, (Long and Short Papers). pp. 4171–4186.
9. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V.J. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
10. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. In Proceedings of the 34th Conference on Neural Information Processing Systems, Online, 6–12 December 2020; pp. 1877–1901.
11. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H.J. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.
12. Su, Y.S.; Han, X.; Lin, Y.K.; Zhang, Z.Y.; Liu, Z.Y.; Li, P.; Zhou, J.; Sun, M.S. CSS-LM: A Contrastive Framework for Semi-Supervised Fine-Tuning of Pre-Trained Language Models. *IEEE ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 2930–2941. [[CrossRef](#)]
13. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J.J. GPT understands, too. *arXiv* **2021**, arXiv:2103.10385. [[CrossRef](#)]
14. Gao, T.; Fisch, A.; Chen, D.J. 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, Online, 1–6 August 2021; pp. 3816–3830.
15. Schick, T.; Schütze, H.J. It’s not just size that matters: Small language models are also few-shot learners. *arXiv* **2020**, arXiv:2009.07118.
16. Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Wang, J.; Li, J.; Wu, W.; Sun, M.J. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 2225–2240.
17. Sun, Y.; Wang, S.; Feng, S.; Ding, S.; Pang, C.; Shang, J.; Liu, J.; Chen, X.; Zhao, Y.; Lu, Y.J. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv* **2021**, arXiv:2107.02137.
18. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. In Proceedings of the 34th Conference on Neural Information Processing Systems, Online, 6–12 December 2020; pp. 18661–18673.
19. Dan, Y.; Zhou, J.; Chen, Q.; Bai, Q.; He, L. Enhancing class understanding via prompt-tuning for zero-shot text classification. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 4303–4307.
20. Zhang, W.; Song, X.; Feng, Z.; Xu, T.; Wu, X.J. LabelPrompt: Effective Prompt-based Learning for Relation Classification. *arXiv* **2023**, arXiv:2302.08068.
21. You, Y.; Jiang, Z.; Zhang, K.; Jiang, J.; Wang, X.; Zhang, Z.; Wang, S.; Feng, H. TI-Prompt: Towards a Prompt Tuning Method for Few-shot Threat Intelligence Twitter Classification. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, 27 June–1 July 2022; pp. 272–279.
22. Zhang, H.; Liang, B.; Yang, M.; Wang, H.; Xu, R.F. Prompt-Based Prototypical Framework for Continual Relation Extraction. *IEEE ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2801–2813. [[CrossRef](#)]
23. Shin, T.; Razeghi, Y.; Logan IV, R.L.; Wallace, E.; Singh, S.J. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–18 November 2020; pp. 4222–4235.
24. Schick, T.; Schütze, H.J. 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.
25. Jiang, T.; Jiao, J.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Deng, D.; Zhang, Q.J. Promptbert: Improving bert sentence embeddings with prompts. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 8826–8837.
26. Gao, T.; Yao, X.; Chen, D.J. Simcse: Simple contrastive learning of sentence embeddings. *arXiv* **2021**, arXiv:2104.08821.
27. Guo, J.; Zhao, B.; Liu, H.; Liu, Y.; Zhong, Q.J.T.S. Technology Supervised contrastive learning with term weighting for improving Chinese text classification. *Tsinghua Sci. Technol.* **2022**, *28*, 59–68. [[CrossRef](#)]
28. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 2901–2908.
29. Liu, J.; Zhang, Z.; Guo, Z.; Jin, L.; Li, X.; Wei, K.; Sun, X.J.K.-B.S. KEPT: Knowledge Enhanced Prompt Tuning for event causality identification. *Knowl. Based Syst.* **2023**, *259*, 110064. [[CrossRef](#)]
30. Song, C.; Cai, F.; Wang, M.; Zheng, J.; Shao, T.J.K.-B.S. TaxonPrompt: Taxonomy-aware curriculum prompt learning for few-shot event classification. *Knowl. Based Syst.* **2023**, *264*, 110290. [[CrossRef](#)]

31. Qi, F.; Zhang, L.; Yang, Y.; Liu, Z.; Sun, M. Wantwords: An open-source online reverse dictionary system. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 175–181.
32. Xu, B.; Xu, Y.; Liang, J.; Xie, C.; Liang, B.; Cui, W.; Xiao, Y. CN-DBpedia: A never-ending Chinese knowledge extraction system. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Online, 27–30 June 2017; pp. 428–438.
33. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
34. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R.J. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
35. Xu, L.; Lu, X.; Yuan, C.; Zhang, X.; Xu, H.; Yuan, H.; Wei, G.; Pan, X.; Tian, X.; Qin, L.J. Fewclue: A Chinese few-shot learning evaluation benchmark. *arXiv* **2021**, arXiv:2107.07498.
36. Chen, Q.; Zhang, R.; Zheng, Y.; Mao, Y.J. Dual contrastive learning: Text classification via label-aware data augmentation. *arXiv* **2022**, arXiv:2201.08702.
37. Hambardzumyan, K.; Khachatrian, H.; May, J.J. Warp: Word-level adversarial reprogramming. In Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; pp. 4921–4933.

**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.