Complicate then Simplify: A Novel Way to Explore Pre-trained Models for Text Classification

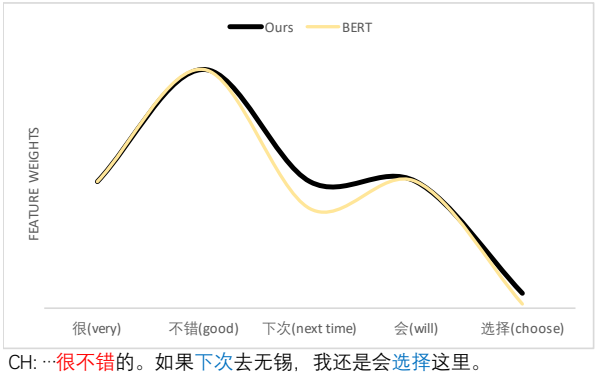
Xu Zhang and **Zejie Liu** and **Yanzheng Xiang** and **Deyu Zhou***
School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
{xuzhang123,liuzejie,yz_xiang,d.zhou}@seu.edu.cn

Abstract

In the developing context of pre-trained models (PTMs), the performance of text classification has been continuously improved by directly employing the features generated by PTMs. However, such a way might not fully explore the knowledge in PTMs as it is constrained by the difficulty of the task. Compared to a difficult task, the learning algorithms tend to saturate early on the simple task. Moreover, the native sentence representations derived from BERT are prone to be collapsed and directly employing such representation for text classification might fail to fully capture discriminative features. In order to address these issues, in this paper we propose a novel framework for text classification which implements a two-stage training strategy. In the pre-training stage, auxiliary labels are introduced to increase the task difficulties and to fully exploit the knowledge in the pre-trained model. In the fine-tuning stage, the textual representation learned in the pre-training stage is employed and the classifier is fine-tuned to obtain better classification performance. Experiments were conducted on six text classification corpora and the results showed that the proposed framework outperformed several state-of-the-art baselines.

1 Introduction

Text classification is a fundamental task in the field of natural language processing and is widely employed in various tasks such as question answering, sentiment analysis, and information retrieval. With the continuous development of machine learning algorithms, especially the success of deep learning methods, text classification has been significantly improved, e.g. CNNs (Kim, 2014; Lai et al., 2019), RNNs (Chen et al., 2017; Zhang et al., 2020), BERTs (Cui et al., 2019, 2020; Sun et al., 2021), etc. Recently, pre-trained models have been shining in classification-based natural language processing tasks (Cui et al., 2019, 2020; Sun et al., 2021).



CH: …很不错的。如果下次去无锡，我还是会选择这里。
EN: …were very good. If I go to Wuxi next time, I would still choose this place.

Figure 1: An Example from the ChnSentiCorp corpus.

The advent of BERT has led to an effective enhancement of textual feature representation. A series of improved pre-trained models have been proposed, e.g. RoBERTa (Liu et al., 2019), MacBERT (Cui et al., 2020), ERNIE (Sun et al., 2020), ChineseBERT (Sun et al., 2021). For example, ERNIE learns real-world semantic knowledge by modeling words, entities and entity relationships in massive amounts of data (Sun et al., 2020). Sun et al. (2021) proposed a large-scale Chinese pre-trained model that incorporates glyph and pinyin information. The above works improve the textual representation of the pre-trained model by introducing external knowledge, without fully exploring the semantic representation in the existing pre-trained model. The recent work, prompt learning, is the technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input (Liu et al., 2021). Like prompt learning, we propose to further extract more meaningful textual representations from the pre-trained model, i.e. to make the extracted textual semantic representation more discriminative for classification.

Although the current pre-trained model already obtain relatively good textual representation, it is

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 1136–1145
October 12–17, 2022.

still possible to further explore the information in the pre-trained representation based on our observation. As shown in Figure 1, for sentiment analysis, there are two sets of features in a sentence that indicate its positive emotional polarity, “很不错(very good)” and “下次 选择(next time choose)”. Words like “很 不错(very good)” explicitly express positive sentiment. Features like these, which are closely related to category labels, are given higher weight during training. While more implicit features like “下次(next time)” and “选择(choose)” are easily ignored as shown by the green curve in Figure 1. However, by using the two-stage framework proposed in this paper, the weights of implicit discriminative features are further highlighted without weakening the weights of the most discriminative feature as shown in the black curve in Figure 1.

Meanwhile, we notice that Yan et al. (2021) found the word representation space of BERT to be anisotropic, with high-frequency words clustered together and close to the origin, while low-frequency words were sparsely scattered. When averaging token embeddings, those high-frequency words dominate the sentence representation, inducing a bias against their actual semantics. Such phenomenon has also been observed in some previous work (Gao et al., 2019; Wang et al., 2020a; Li et al., 2020). Therefore, directly employing such representation for text classification might fail to fully capture discriminative features.

Therefore, in this paper, we consider the extracting of semantic features from a cognitive perspective by introducing auxiliary labels and constructing pre-training and fine-tuning strategies based on pre-trained models. We devise a novel approach to perform a secondary pre-training based on the pre-trained model and then fine-tuning for text classification which is similar to the process of gaining new insights through restudying old material. In the pre-training stage, the model learns a better representation of the task under consideration. In the fine-tuning stage, the classifier is fine-tuned by applying the text feature representations obtained from the pre-training stage. To fully exploit the discriminative features in the pre-trained model, in the pre-training stage, auxiliary labels are constructed to take fine-grained semantic categories into account. The introduction of auxiliary labels makes the information entropy increase. Knowledge in the pre-trained model is fully mined for a more effective discriminative semantic representation.

The main contributions are listed as follows.

- We propose a novel framework for text classification which implements a two-stage training strategy and enables "experience accumulation" and "practice what you learned" without introducing additional knowledge.
- In the pre-training stage, auxiliary labels are integrated to increase the training challenge and to exploit the knowledge in the pre-trained model fully.
- The validity of the framework is verified on seven benchmark datasets, and the proposed framework achieves better performance than several state-of-the-art baselines.

The reminder of the paper is structured as follows. Some related work is briefly reviewed in Section 2. The detailed implementation of our framework is described in Section 3. Section 4 reports the experiments and results and Section 5 shows further analysis and discussion. Finally, the paper is concluded in Section 6.

2 Related Work

The development of deep learning has led to significant improvements in text classification, and some of the more widely employed deep learning models for text classification tasks are CNNs (Wang et al., 2018; Lai et al., 2019), RNNs (Chen et al., 2017; Sachan et al., 2019; Zhang et al., 2020), and pre-trained models (Cui et al., 2019; Liu et al., 2019; Sun et al., 2021). In recent years, pre-trained models have shown excellent performance in the field of text classification. Whether CNNs, RNNs, or more recently pre-trained models, the purpose of employing deep models is to efficiently capture the textual semantic representation.

2.1 Traditional Deep Learning methods

Text representation is the basis for text classification. The first work to introduce CNNs into NLP was done by Kim (2014), and the key to the features captured by a CNN is the sliding window covered by the convolutional kernel. Johnson and Zhang (2017) proposed the Deep Pyramidal Convolutional Neural Network (DPCNN), which can effectively extract remote relational features from the text. In the process of text feature extraction employing convolutional operations, the semantic relations of sentences would be lost. Ma et al. (2015)

proposed to employ dependent syntactic trees to extract the semantic feature relations, instead of just employing adjacent word representations as feature representations. CNNs can extract local features from global information when employed for text classification, but they are unable to capture long-term dependencies, whereas RNNs can. Zhang et al. (2018) proposed a sentence-state based LSTM that incorporates the semantic relevance of words and sentences. Models such as CNNs and LSTMs capture word sense information well in locally continuous word sequences but may ignore global word co-occurrences in corpora with discontinuous and long-term semantics dependencies. Yao et al. (2019) proposed a graph-based convolutional neural network structure (GCN), which exploits global word co-occurrence information not previously considered by other models and demonstrates better robustness with less training data. RNNs suffer from gradient explosion and gradient disappearance, and cannot effectively handle long-term context-dependence. The attention mechanism can characterize the target location by linearly weighting the features of the contextual source sequence. Bahdanau et al. (2015) first applied attention mechanisms to the field of natural language processing. Yang et al. (2016) proposed a hierarchical attention mechanism model for text classification tasks, acting at the word and sentence levels respectively. With the development of deep learning, neural networks are widely employed in NLP tasks, such as the aforementioned CNNs, RNNs, GNNs and attention mechanisms, but as the available datasets are small for most supervised NLP tasks, the above models are "shallow" for NLP tasks, making it difficult to extract sufficiently rich textual representations.

2.2 Pre-trained Models

The advent of pre-trained models (PTMs) has ushered in a new era of NLP, with extensive work showing that pre-trained models on large corpora can learn generic language representations and avoid training from scratch when solving downstream NLP tasks (Liu et al., 2019; Sun et al., 2020, 2021). Since BERT, complementary pre-trained models have been designed to integrate external knowledge into PTMs for better textual representations. ERNIE combines pre-trained entity embeddings in the knowledge graph with corresponding entity mentions in text to enhance the text representation (Zhang et al., 2019). KnowBERT merges entity representations in an end-to-end manner (Peters et al., 2019). KEPLER unites knowledge embeddings with language model objects (Wang et al., 2021). K-BERT differs from the above models by introducing structured information from the knowledge graph through entity embeddings (Liu et al., 2020). It obtains an expanded tree input to the BERT by directly introducing relevant triples from the knowledge graph into the sentence. K-Adapter independently trains different adapters for different pre-trained models to introduce multiple knowledge, in order to address the forgetting problem that occurs when the above models are injected with multiple knowledge (Wang et al., 2020b). In contrast to the above approach, Qin et al. (2020) proposed the use of feature projection methods based on BERT to further improve text representation without introducing external knowledge. They consider fully mining the existing knowledge in the pre-trained model to make the feature representations involved in classification more discriminative.

3 Model

The overall framework is shown in Figure 2. The whole framework is divided into two stages: pre-training and fine-tuning. In the pre-training stage, auxiliary labels are introduced to artificially boost the training difficulty, which can better tap the knowledge from pre-trained models and obtain a more discriminative textual feature representation. In the fine-tuning stage, the textual representation pre-trained in the pre-training stage is employed and the classifier is fine-tuned to obtain better classification performance.

3.1 Problem Definition

Suppose that we have a K-class classification task, a training instance can be denoted as (x_i, y_i) for $i = 1, ..., N$ and $y_i \in \{1, 2, \ldots, K\}$. Here, we introduce auxiliary labels, as shown in Figure 2. Suppose we have an encoder $E(\cdot)$.

$$R = \mathrm{E}(x) \tag{1}$$

Firstly, the model is trained employing auxiliary labels to obtain discriminative textual representation R. Afterwards the classification is performed by the textual representation R, and the original label y_i.

Auxiliary Labels: The auxiliary labels are introduced through expanding the target labels by com-

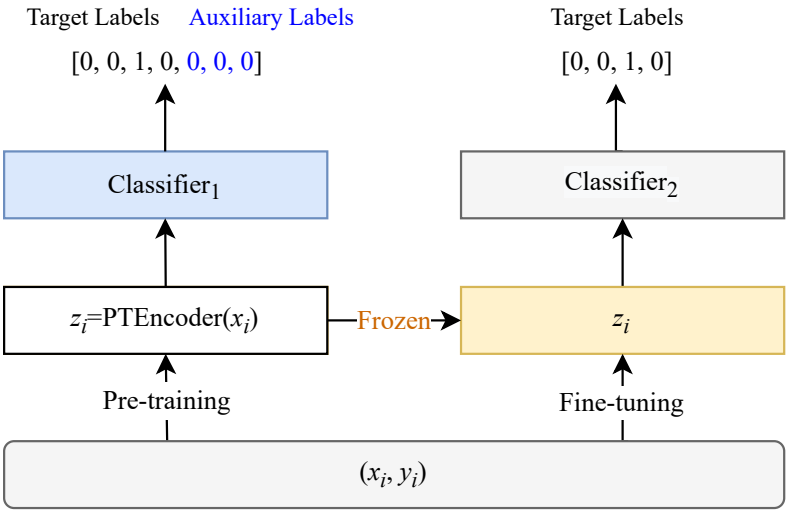


Figure 2: Architecture of the proposed approach.

plementary zeros, simulating the existence of fine-grained label categories. During the pre-training stage, the auxiliary labels also have predicted probability values. Therefore the probability values of the predicted labels corresponding to the target labels fluctuate and the loss function decreases at a slower rate during pre-training. To gain better classification results, the model is forced to learn a better textual representation.

Information Entropy: Suppose that, for a K classification task in which the prediction probabilities of the false labels are all γ, the prediction probability of the true label is (1 - (k -1)γ). The information entropy is shown below.

$$H_k = -\sum_{i=1}^{k-1} \gamma \log \gamma - (1-(k-1)*\gamma) \log(1-(k-1)*\gamma) \quad (2)$$

Similarly, for the case of introducing n auxiliary labels, the information entropy is as follows:

$$H_{k+n} = -\sum_{i=1}^{k+n-1} \gamma \log \gamma - (1-(k+n-1)*\gamma) \log(1-(k+n-1)*\gamma) \quad (3)$$

It can be inferred that the introduction of auxiliary labels makes the information entropy increase as shown in Equation 4. The lower the deviation of H_{k+n} and H_k, the smaller the difference between the two distributions. After introducing the auxiliary labels, the model needs to mine the knowledge in the pre-trained model in-depth, in order to reduce the gap with the true distribution.

$$H_{k+n} - H_k > 0 \quad (4)$$

3.2 Pre-training Pre-trained Models and Fine-tuning

The traditional text classification strategy is to obtain the textual semantic representation through the Encoder Network $Encoder(\cdot)$ and then employ the Classifier $Classifier(\cdot)$ to make predictions.

$$z_i = \text{Encoder}(x_i) \quad (5)$$

$$c_i = \text{Classifier}(z_i) \quad (6)$$

θ is the parameter of $Encoder(\cdot)$ and β is the parameter of $Classifier(\cdot)$. Their method is described as follows:

$$\mathscr{L}(\theta,\beta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} 1(y_i = k) \log(k \mid c_i) \quad (7)$$

Unlike the traditional classification models described above, our approach adopts a two-stage training strategy.

Pre-training : Similarly, the input text x_i needs to be represented as a textual semantic representation employing a parameters trainable encoder $PTEncoder(\cdot)$. Here, one point to focus on is that the parameters of the encoder are trainable.

$$z_i = \text{PTEncoder}(x_i) \quad (8)$$

$PTEncoder(\cdot)$, which maps x_i to a discriminative representation vector shown in Equation 8.

Next, label prediction is to be performed employing z_i. Although the auxiliary labels are introduced, this is only done to introduce interference terms, in order for the $PTEncoder(\cdot)$ to be trained to obtain

more discriminative text features for the subsequent classification.

$$c1_i = \text{Classifier}_1(z_i) \tag{9}$$

θ_1 is the parameter of $PTEncoder(\cdot)$ and β_1 is the parameter of $Classifier_1(\cdot)$. Their method is described as follows:

$$\mathscr{L}_\infty(\theta', \beta') = -\sum_{i=1}^{N}\sum_{k=1}^{K+j} 1\,(y1_i = k)\log\,(k \mid c1_i) \tag{10}$$

where j is the number of auxiliary labels we introduce on top of the original K targets.

In addition, there is no other effect on the classification task, as the auxiliary labels are represented in the one-hot labels as 0. Therefore, in the specific task, only the original categories are involved in the calculation of the cross-entropy loss function, and the auxiliary labels are not involved.

Fine-tuning : With the first stage (Pre-training), better parameters for the $PTEncoder(\cdot)$ can be pre-trained, by which a better discriminative textual representation z_i can be obtained.

Next, the trained textual representation z_i is employed for conventional classification, i.e., no auxiliary labels are introduced and classification is performed according to the given label categories.

Here, we only perform further fine-tuning for the classifier:

$$c2_i = \text{Classifier}_2(z_i) \tag{11}$$

θ_1 is the parameter of $PTEncoder(\cdot)$ and β_2 is the parameter of $Classifier_2(\cdot)$. Their method is described as follows:

$$\mathscr{L}_\in(\theta_1, \beta_2) = -\sum_{n=1}^{N}\sum_{k=1}^{K} 1\,(y_i = k)\log\,(k \mid c2_i) \tag{12}$$

A summary of the two-stage training strategy described above. (1) In the pre-training stage, auxiliary labels for classification are constructed and the encoder is pre-trained employing the auxiliary labels to obtain a more discriminative feature representation z_i. (2) In the fine-tuning stage, the textual representation z_i is employed for classification, where only fine-tuning is done for the classifier and the parameters of the text representation z_i are no longer updated to obtain better classification performance.

3.3 Textual Representation

BERT (Devlin et al., 2019) is a multilayered attention-assisted bidirectional transformer encoder model based on the original transformer model (Vaswani et al., 2017). During pre-training, BERT employed two objectives: masked language model (MLM) and next sentence prediction (NSP). The NSP is employed to predict whether two segments follow each other. The goal of NSP is to improve the performance of downstream tasks such as natural language inference (Bowman et al., 2015), which entails reasoning about the relationship between pairs of sentences. NSP is a good fit with our matching-based QA task and the matching task. Therefore, we choose the BERT as the encoding model.

The original authors of BERT proposed an upgraded version of BERT, including Whole Word Masking (WWM), which alleviates the disadvantage of masking some WordPiece tokens in pre-trained BERT. Cui et al. employed the whole word masking strategy for Chinese BERT and published a series of Chinese pre-trained models (Cui et al., 2019). The experimental performance shows that the proposed pre-trained model yields substantial improvements over BERT and ERNIE on various NLP tasks. They adapted whole-word masking in Chinese text by masking whole words instead of Chinese characters.

In view of the excellent performance achieved by BERT-wwm in Chinese tasks (Cui et al., 2019), an improved version of the BERT model proposed by Cui et al. is employed as the Encoder in our work.

4 Experiment

4.1 Datasets

We conducted experiments on six Chinese text classification datasets, including two sentence semantic matching datasets (BQ (Chen et al., 2018) and LCQMC (Liu et al., 2018)), one text classification datasets (TNEWS (Xu et al., 2020)), one sentiment classification dataset (ChnSentiCorp [1]), and two Chinese question answering datasets from the NLPCC-2016 evaluation task (Duan, 2016).

Sentence semantic matching task: BQ is the largest Chinese question matching dataset in the banking domain. LCQMC is the largest Chinese

[1] https://github.com/pengming617/bert_classification/tree/master/data

semantic matching dataset available and obtained from the Baidu Knows question and answer community.

Text classification task: TNEWS selects from the news section of Today's Headlines, with 15 news categories, including travel, education, finance, military and more. ChnSentiCorp is a Chinese sentiment classification dataset with more than 7000 hotel reviews, 5000 positive reviews and 2000 negative reviews

Question answering task: DBQA is a document-based question answering dataset. These candidate sentences were extracted from web pages and tended to be much longer than the questions, with many irrelevant sentences. KBQA is a knowledge base based question answering dataset, and each question contained only one golden predicate.

4.2 Baselines

The following approaches are employed as the baselines, including BERT-wwm and BERT-wwm-ext (employing extended data, including Chinese Wikipedia, other encyclopedias, news, QAs and other data, with a total word count of 5.4B) (Cui et al., 2019). To evaluate the proposed approach, we additionally selected a series of pre-trained models for comparison, with RoBERTa-base, RoBERTa-large (Liu et al., 2019), MacBERT-base, MacBERT-large (Cui et al., 2020), ChineseBERT-base, and ChineseBERT-large (Sun et al., 2021).

Our approach is customized to several versions to evaluate its performance, as follows: We improve on the BERT-wwm, BERT-wwm-ext by introducing our approach to obtain RE-BERT-wwm, RE-BERT-wwm-ext. MRE-BERT fuses the textual representations of the BERT-wwm and the BERT-wwm-ext models and then introduces our approach for optimization.

4.3 Experiment Setup

The experimental setup is shown in Table 1. We found experimentally that the number of auxiliary labels is set differently for different pre-trained models and various tasks. As a hyperparameter, it needs to be adapted to the different training tasks. For LCQMC and TNEWS datasets with relatively clear classification goals and little ambiguity in the annotated data, introducing auxiliary labels did not significantly improve classification but the two-stage repetitive operation based on learning similar to human cognitive skills still improved the model's effectiveness. In addition, the batch size is set to 64, and Adam with parameters 2e-5 is employed as the optimizer (Kingma and Ba, 2015). All experiments were performed on two Nvidia Tesla T4 GPUs.

4.4 Experiment Results

4.4.1 Matching based QA Task

As shown in Table 2, following the work of lai et al. (Lai et al., 2019), we have implemented BERT-based Chinese KBQA and DBQA tasks based on their work. In their work, Lattice CNNs were employed as encoder, and we experimented with BERTs-base replacing Lattice CNNs as the baseline for our work.

As shown in Table 2, we have improved the experiments by employing our approach compared to the original BERT-wwm and BERT-wwm-ext. The experiments show that training through two stages is effective and our proposed approach of introducing auxiliary labels In the pre-training stage is feasible. The introduction of auxiliary labels allows for more meaningful discriminative features for classification in feature extraction.

M-BERT unites the textual semantic representations of two pre-trained models, BERT-wwm and BERT-wwm-ext. Compared to employing a single pre-trained model, BERT-wwm or BERT-wwm-ext, there is an improvement in the experimental results. The experiment also validates our previous consideration that there is variability in the textual semantic representation obtained by different pre-trained models and that an improvement can be achieved by combining multiple pre-trained models.

MRE-BERT has a significant improvement over the two BERT-base models, BERT-wwm and BERT-wwm-ext, on both DBQA and KBQA question and answers datasets. Compared to BERT-wwm, MRE-BERT has a more than 1% improvement on both datasets. There is also a substantial effect improvement in each evaluation metric compared to BERT-wwm-ext.

4.4.2 Text Classification Task

In addition, we conducted corresponding experiments on two text classification datasets, TNEWS as well as ChnSentiCorp, which have relatively few category labels.

As shown in Table 3, for the TNEWS and ChnSentiCorp datasets, which have relatively few categories, our proposed approach achieves good performance on both datasets. The experimental results for the TNEWS dataset on BERT-wwm as

Dataset	Scale (train/valid/test)	Model	N^a	N^t	Epoch
ChnSentiCorp	9.6K/1.2K/1.2K	RE-BERT-wwm	4	2	10
		RE-BERT-wwm-ext	4	2	10
		MRE-BERT	16	2	10
TNEWS	12.1k/2.6k/2.6k	RE-BERT-wwm	119	119	6
		RE-BERT-wwm-ext	15	15	3
		MRE-BERT	15	15	3
DBQA	182k/-/123k	RE-BERT-wwm	4	2	2
		RE-BERT-wwm-ext	4	2	2
		MRE-BERT	128	2	2
KBQA	273k/-/156k	RE-BERT-wwm	4	2	2
		RE-BERT-wwm-ext	16	2	2
		MRE-BERT	6	2	2
LCQMC	238.7k/8.8k/12.5k	MRE-BERT	2	2	3
BQ	100k/10k/10k	MRE-BERT	12	2	5

Table 1: Experimental parameter settings. N^a means the whole number of auxiliary and target labels, and N^t number of target labels

	DBQA			KBQA	
Model	**P@1**	**MRR**	**MAP**	**P@1**	**MRR**
BERT	90.06	93.79	93.75	92.88	95.69
RE-BERT	90.20	93.80	93.77	93.80	96.28
BERT^o	90.95	94.38	94.35	93.23	95.90
RE-BERT^o	91.18	94.57	94.54	94.01	96.42
M-BERT	91.08	94.36	94.31	93.75	96.31
MRE-BERT	**91.91**	**95.03**	**95.00**	**94.16**	**96.56**

Table 2: Experimental results on matching based QA task. BERT represents BERT-wwm and o represents models pre-trained on extended data.

well as BERT-wwm-ext are from CLUE[2] (Xu et al., 2020), and the experimental results for ChnSentiCorp are from ChineseBERT (Sun et al., 2021). The experimental results show that MRE-BERT has a better effect compared to the two base pre-trained models BERT-wwm and BERT-www-ext.

	TNEWS	ChnSentiCorp
Model	**Valid**	**Test**
BERT	56.09	95.4
BERT^o	56.77	95.3
RE-BERT	56.87	94.8
RE-BERT^o	**57.40**	95.6
MRE-BERT	56.95	**95.8**

Table 3: Experimental on text classification task. BERT represents BERT-wwm and o represents models pre-trained on extended data.

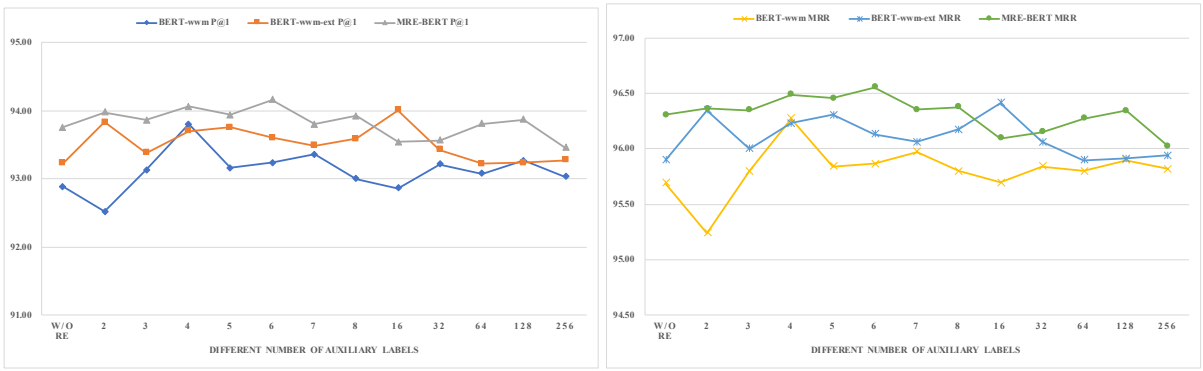
[2]https://github.com/CLUEbenchmark/CLUE

5 Analysis and Discussion

5.1 Comparison of Different Auxiliary Labels

In order to verify the effectiveness of our approach and the effect of the auxiliary labels, we conducted some targeted experiments on the KBQA dataset.

As shown in Figure 3, we have selected a series of samples for classification after adding auxiliary labels In the pre-training stage (pre-training) for RE-BERT-wwm, BERT-wwm-ext and MRE-BERT models respectively. The w/o RE model shows the effect without our approach, from which it can be seen that models can be effectively improved by introducing auxiliary labels. With the experimental results in Figure 3, we have verified the effectiveness of our approach. The introduction of auxiliary labels helps to extract textual semantic representations efficiently, and the number of auxiliary labels needs to be set according to the needs of different models and tasks.



(a) Comparison of the effectiveness on the P@1.

(b) Comparison of the effectiveness on the MRR.

Figure 3: Comparison of the effects of different models on KBQA dataset after the addition of auxiliary labels In the pre-training stage.

5.2 Verification Experiments

Model	BQ		LCQMC	
	Valid	**Test**	**Valid**	**Test**
	base			
BERT	86.1	85.2	89.4	87.0
BERT^o	86.4	**85.3**	89.6	87.1
RoBERTa^o	86.0	85.0	89.0	86.4
MacBERT	86.0	85.2	89.5	87.0
ChineseBERT	86.4	85.2	89.8	**87.4**
MRE-BERT	86.3	**85.6**	90.4	**87.4**
	large			
RoBERTa^o	86.3	85.8	90.4	87.0
MacBERT	86.2	85.6	90.6	87.6
ChineseBERT	86.5	86.0	90.5	**87.8**
	base			
MRE-BERT$_F$	86.7	**86.2**	89.9	**87.7**

Table 4: Validation experiments on text matching task. BERT represents BERT-wwm, o represents models pre-trained on extended data, and MRE-BERT$_F$ represents MRE-BERT with adversarial perturbation.

To further validate the effectiveness of MRE-BERT, some related experiments are conducted on two sentence semantic matching datasets and compared with current state-of-the-art pre-trained models. The experimental results show that MRE-BERT has achieved better performance compared to BERT-wwm, BERT-wwm-ext, RoBERTa-wwm-ext (Cui et al., 2019), MacBERT-base (Cui et al., 2020), and ChineseBERT-base (Sun et al., 2021). Furthermore, adding random perturbations to its embedding layer based on pre-trained models is effective in many tasks, expanding the training data by perturbing the samples and being able to regularise the model effectively (Ju et al., 2019). We also tried to introduce adversarial training In the pre-training stage, and the performance was further improved. We notice that ChineseBERT considers two important aspects specific to Chinese: glyphs and pinyin, which carry important syntactic and semantic information for language understanding. It incorporates both glyph and pinyin information into pre-trained language models, achieving new SOTA performance in a wide range of Chinese NLP tasks. However, our proposed BERT-base based framework achieves comparable results to ChineseBERT.

6 Conclusion

For simple text classification tasks, we propose a novel framework for simple text classification tasks, which implements a two-stage training strategy, including pre-training based on pre-trained models and fine-tuning for classification. In the pre-training stage, auxiliary labels can be integrated to increase the training challenge and to fully exploit the knowledge in the pre-trained model. Experiments on six datasets depict that our approach outperforms the baseline BERT model for simple classification tasks. Furthermore, in the sentence semantic matching task, after adding adversarial perturbations to the embedding layer only, our basic version of the MRE-BERT model achieves promising performance, better than or equivalent to large versions of the pre-trained model, but with much fewer parameters than them. In the future, We will investigate the generalizability of our model to other classification tasks.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments. This work was funded by the National Natural Science Foundation of China (62176053).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4946–4951.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

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

Nan Duan. 2016. Overview of the nlpcc-iccpol 2016 shared task: open domain chinese question answering. In *Natural Language Understanding and Intelligent Applications*, pages 942–948. Springer.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Yuxuan Lai, Yansong Feng, Xiaohan Yu, Zheng Wang, Kun Xu, and Dongyan Zhao. 2019. Lattice cnns for matching based chinese question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6634–6641.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mingbo Ma, Liang Huang, Bowen Zhou, and Bing Xiang. 2015. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings*

of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 174–179.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.

Qi Qin, Wenpeng Hu, and Bing Liu. 2020. Feature projection for improved text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8161–8171.

Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6940–6948.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2065–2075.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020a. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. 2020b. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018. Densely connected cnn with multi-scale feature attention for text classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4468–4474.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Xu Zhang, Yifeng Li, Wenpeng Lu, Ping Jian, and Guoqiang Zhang. 2020. Intra-correlation encoding for chinese sentence intention matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5193–5204.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.