Integrating R-Drop and Pre-trained Language Model for Short Text Classification

1st Dengfeng Liu
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
liudengfeng21@nudt.edu.cn

2nd Fei Cai *
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
caifei08@nudt.edu.cn

3rd Zhiqiang Pan
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
panzhiqiang@nudt.edu.cn

4th Jianming Zheng
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
zhengjianming12@nudt.edu.cn

5th Yanying Mao
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
maoyanying@nudt.edu.cn

6th Mengru Wang
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
wangmengru20@nudt.edu.cn

2022 8th International Conference on Big Data and Information Analytics (BigDIA) | 978-1-6654-8796-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIGDIA56350.2022.9874173

***Abstract*—At present, the rapid development of computer technology has led to the emergence of massive short text data represented by e-commerce user comments. Due to the sparse feature and fast update speed of short text data, effective classification methods of short text have been widely concerned and applied. In this paper, a short text classification method integrating R-Drop and pre-trained language model IRDP-STC is proposed. The original data and generated data are input into the pre-trained language model Bert with Dropout, and KL divergence loss is added to approximate the two outputs. IRDP-STC makes full use of short text information, takes into account the robustness of the model on the basis of ensuring the high classification performance of the model, and is more suitable for low training sample scenarios. The experimental results on THUCNews dataset are better than the baseline model, which verifies the effectiveness and universality of the proposed method.**

***Key Words*—R-Drop, Pre-trained language model, Short text classification**

I. Introduction

Text classification is one of the cornerstone tasks in the field of natural language processing (NLP), which is a research hotspot in academia and industry. At present, with the rapid development of Internet industry, the short text data such as e-commerce user comments and news summaries, are massively generated at all time. Based on effective short text classification methods to capture the target information, in sentiment analysis, spam filtering, current affairs hot spot tracking, public opinion monitoring and many other practical scenarios have extensively researched and applied.

* Fei Cai is corresponding author.

Text classification aims to understand the given natural language text by computer and automatically obtain the category of the text. In terms of short text classification and long text classification, the number of text words is rather small. For example, if the traditional method of extracting text word frequency features is used, such as Term Frequency-Inverse Document Frequency (TF-IDF), it is prone to sparse text features and difficult to ensure the classification performance. At the same time, due to the rapid update of short text data, if the classification model is sensitive to the classification data and the robustness is not strong, it is difficult to guarantee the classification performance of new data.

To solve the above problems, this paper proposes an Integrating R-Drop and Pre-trained language model for Short Text Classification (IRDP-STC), which integrates R-Drop [1] and pre-trained language model. Specifically, with the help of the rich prior knowledge of pre-trained language model Bidirectional Encoder Representation from Transformers (Bert), better short text semantic features are captured, which is more suitable for the downstream short text classification tasks. At the same time, the data generator is constructed to allow each training sample to replicate the data and then feed them into the pre-trained language model with Dropout. The outputs of the same input sample is approximated by KL divergence, so as to improve the robustness of the model. This paper has three main contributions:

(1) A short text classification framework integrating R-Drop and pre-trained language model is proposed, which captures rich text semantic information and maintains model robustness in short text classification tasks, ensures the classification performance of the model and can quickly migrate to the new

978-1-6654-8796-2/22/$31.00 © 2022 IEEE

text data scene.

(2) The short text classification framework is adaptable to low-resource scenarios with strong ability and compatibility. In the case of insufficient training corpus, it can still make full use of the existing data set information to ensure the model performance. Moreover, the pre-trained language model and classifier can be replaced according to the actual data scene, so as to be applicable to the required application scene.

(3) The effectiveness of the proposed method is verified by THUCNews dataset.

II. RELATED WORK

The current text classification methods can be roughly divided into three categories.

A. Text Classification Method Based on Knowledge Engineering

Knowledge engineering based classification method is very simple. The classification is completed by making expert system or counting the word frequency of keywords in the to be classified text. Luhn et al. counted the frequency of keywords in the title and abstract to classify the text. Although this method is clear and simple, the classification result is poor, and the matching rules made by experts are expensive and inefficient.

B. Text Classification Method Based on Machine Learning

The processing paradigm of text classification method based on machine learning usually first preprocesses text data, then extracts text features such as bag of words model, N-Gram, TF-IDF, and finally trains the classifier through the corresponding machine learning algorithm. Joachims introduced SVM into text classification for the first time. Since then, many machine learning algorithms had been applied, such as K-nearest neighbor, Naive Bayes, decision tree and so on. The traditional machine learning algorithms usually get high-dimensional and sparse text vectors, and do not consider the semantic relationship between words and sentences. The feature representation ability is weak and the generalization is poor.

C. Text Classification Method Based on Deep Learning

After Hinton et al. proposed the concept of deep learning, the text classification algorithm based on deep learning has become the current mainstream research work. Literature [2] introduced convolutional neural network into text classification for the first time, and captured the local correlation of text by the way similar to N-Gram, which surpassed the effects of many traditional text classification methods. Literature [3] applied RNN to construct language models. In view of the poor performance of RNN for long texts, Zhang Y et al. integrated the semantic correlation between words and sentences into LSTM to improve the model performance. To learn context information at the same time, BiLSTM is applied to text classification in literature [4]. In addition to the single deep learning model, it is difficult to improve the classification performance due to the lack of text information captured by single model. There are many different hybrid models. The representative model, such as C-LSTM which input the text features extracted by CNN into LSTM.

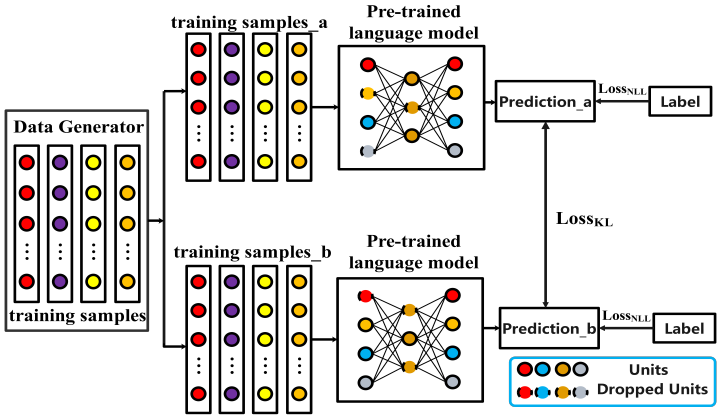

Fig. 1. IRDP-STC Model framework.

The core idea of Attention Mechanism is to capture higher-value information based on limited attention. Bahdanau D et al. first applied attention mechanism to machine translation, and attention mechanism has become one of the hotspots in NLP research. Literature [5] proposed the self-attention mechanism, and the constructed transformer structure abandoned the classical structures such as RNN, GRU and LSTM that were commonly used before. Google proposed Bert based on Transformer structure, which not only has excellent results in many NLP fields [7], but also makes NLP research appear a new paradigm based on pre-trained model. The pre-trained model paradigm usually follows two steps. Firstly, pre-trained model parameters are trained in large corpus. Then, the fine tune based on a specific downstream task dataset. In addition to Bert, there are a series of pre-trained models based on improved transformer structure, such as XLNet and ERNIE.

III. APPROACH

A. Overview of Models

This paper proposes a short text classification framework IRDP-STC integrates R-Drop and pre-trained language model. The overall framework of the model is shown in Fig.1, which is mainly composed of data generation module, integrating R-Drop training module and prediction module.

(1) Data generation module

The data generation module aims to copy each original data sample. The original data and replicated data are input into the same training batch, so as to prepare the required input data of the integrating R-Drop training module.

(2) Integrating R-Drop training module

The integrating R-Drop training module aims to let the original data and the replicated data generated by the data generator pass the pre-trained language model with Dropout respectively. In addition to the traditional cross entropy loss,

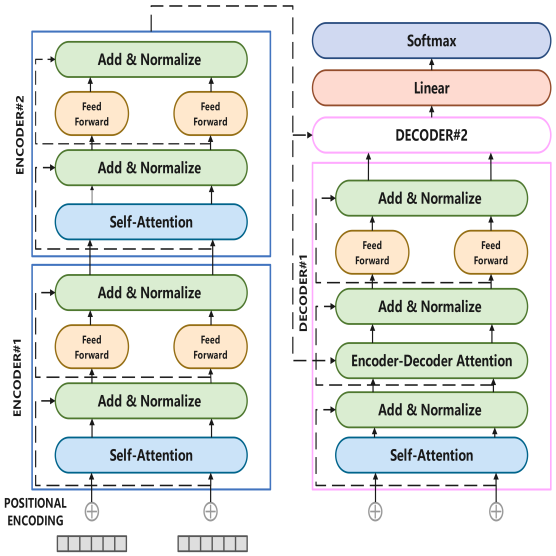

Fig. 2. Transformer internal structure diagram.

KL divergence is added to approximate the outputs to improve the robustness of the model.

(3) Prediction module

The prediction module aims to obtain the text representation of the test data through the pre-trained model trained by the downstream task, and obtain the category of the test data through the classifier.

B. Model Theory and Model Construction

(1) Transformer

The core of the transformer structure is self-attention mechanism, and its calculation formula is shown in formula (1).

$$\text{Attention}(Q, K, V) = \text{Soft max}\left(QK^T d_k^{-\frac{1}{2}}\right)V \quad (1)$$

Where Q, K and V represent the query vector, key vector and value vector corresponding to all words in the input text, respectively. d_k represents the dimension value of query vector and key vector. The internal is composed of Encoder and Decoder. Its structure is different from the classical recurrent neural network and convolutional neural network. Encoder is encoded by multi-layer Encoder, and the structure of each layer is consistent but does not share weight. Decoding operations are made up of multilayer Decoder.

At the same time, the self-Attention layer is often improved by multi-head attention mechanism in practice, so that the model can greatly enhance the location capture range and increase the representation space.

(2) Bert

Bert consists of multiple layers of transformer, the structure of which is schematic in Fig.3.

The Bert input vector can be divided into three parts: position vector, segment vector and word vector, which respectively represent the position of the sentence, subordinate to the upper sentence or the lower sentence, and word meaning, this is shown in Fig.4.

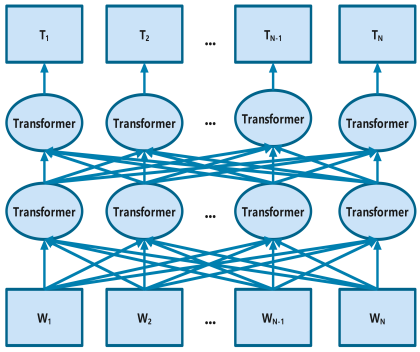

Fig. 3. Structure diagram of Bert model.

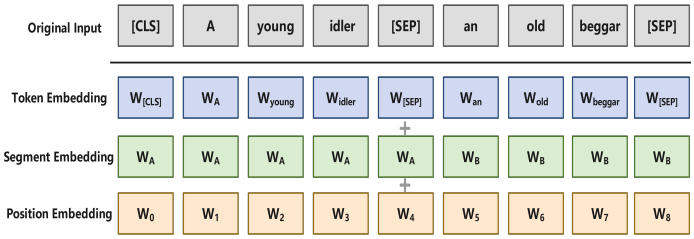

Fig. 4. Bert input schematic.

The formula for calculating the position vector is shown in formula (2) and formula (3).

$$PE(pos, 2i) = \sin\left(\text{pos}/10000^{2i/d}\right) \quad (2)$$

$$PE(pos, 2i+1) = \cos\left(pos/10000^{2i/d}\right) \quad (3)$$

Where, pos represents the position of the word to be encoded and d represents the vector dimension of the word. The word vector includes some special markers such as the sentence begining symbol [CLS]; the sentence segment or ending symbol [Sep]; the sentence length padding symbol [PAD]; and the mask symbol [Mask] . In this paper, [CLS] of the last layer of Bert encoder is used as the text representation, and the probabilities of the samples belonging to each category are obtained by softmax function after the full connection layer.

(3) R-Drop

The core idea of R-Drop is to copy each training sample through the same model with Dropout and approximate the outputs by KL divergence. Because Dropout itself has randomness, equivalent to the same input sample through two slightly different models, so as to ensure that the model has stronger anti-interference ability and robustness.

The R-Drop loss function is mainly composed of two parts. The first part is the traditional cross entropy loss, as shown in formula (4).

$$L_{NLL} = -\log P_\theta^{(1)}\left(y_i \mid x_i\right) - \log P_\theta^{(2)}\left(y_i \mid x_i\right) \quad (4)$$

Where, $P_1 = P_\theta^{(1)}(y_i \mid x_i)$ is the output of the original data through the pre-trained language model, and $P_2 = P_\theta^{(2)}(y_i \mid x_i)$ is the output of the replicated data through the pre-trained language model. The loss based on KL divergence between the two is shown in formula (5).

$$L_{KL} = \frac{1}{2}\left[KL(P_1 \mid P_2) + KL(P_2 \mid P_1)\right] \tag{5}$$

The final loss of the model is the weighted sum of the two losses, as shown in formula (6).

$$L = L_{NLL} + \alpha L_{KL} \tag{6}$$

IV. EXPERIMENT AND ANALYSIS

In this section, the IRDP-STC model is experimentally verified and compared with the traditional method.

A. Experimental set up

(1) Experimental data set

This experiment extracts some short news texts from the widely used benchmark dataset THUCNews to verify the performance of the model.

The dataset includes 10 categories, such as game, finance, education and science, and each category is evenly distributed in each dataset. The training set contains 180,000 training samples, the longest text length is 38, the shortest text length is 3, and the average text length is 19.213. The verification set contains 10,000 training samples, the longest text length is 33, the shortest text length is 4, and the average text length is 19.137. The test set contains 10,000 training samples, the longest text length is 32, the shortest text length is 4, and the average text length is 19.180.

(2) Experimental parameter setting

In this experiment, the main experimental parameters are shown in Table 1.

TABLE I
MAIN EXPERIMENTAL PARAMETERS

Parameter name	Parameter value
batch_size	32
dropout	0.3
α	4
num_epoch	10
learning_rate	5e-5

(3) Evaluation index and comparison model

This experiment verifies the effectiveness of the proposed model by comparing the current mainstream baseline models. Baseline model is as follows :

(1) TextCNN aims to capture text local proximity features by convolutional neural networks.

(2) BiLSTM aims to capture contextual text features by cyclic neural networks.

(3) TextRCNN [8] aims to simultaneously aggregate the advantages of convolutional neural network and cyclic neural network, capture local text features and introduce text context information.

(4) DPCNN [9] is based on the defect that TextCNN cannot capture the long-distance dependency between texts through convolution operation. By deepening the network, the long-distance dependency between texts is extracted without increasing too much operation cost.

(5) Transformer captures text features based on self-attention mechanism.

(6)-(10) Bert and Bert are combined with the above classical text classification neural network to complete the downstream text classification task based on the pre-trained language model.

The evaluation indexes of this paper are Precision, Recall, F1 value and Accuracy.

B. Experiment Results

In order to explore the performance difference between the proposed model and the baseline model, the classification effect of each model in THUCNews short text dataset is shown in table 2.

TABLE II
CLASSIFICATION RESULTS OF EACH MODEL ON THE TEST SET

Model	Precision	Recall	F1-score	Accuracy
TextCNN	0.9107	0.9093	0.9096	0.9093
BiLSTM	0.9094	0.9089	0.9089	0.9089
TextRCNN	0.9183	0.9176	0.9177	0.9176
DPCNN	0.9320	0.9316	0.9314	0.9316
Transformer	0.8990	0.8979	0.8979	0.8979
Bert-base	0.9356	0.9352	0.9352	0.9352
Bert-base-TextCNN	0.9318	0.9314	0.9315	0.9314
Bert-base-BiLSTM	0.9357	0.9349	0.9350	0.9349
Bert-base-RCNN	0.9183	0.9176	0.9177	0.9176
Bert-base-DPCNN	0.9320	0.9316	0.9314	0.9316
IRDP-STC	**0.9466**	**0.9462**	**0.9462**	**0.9462**

The experimental results show that DPCNN is the best performance baseline model based on neural network. Based on the pre-trained language, except that Bert-base-BiLSTM is slightly higher than Bert-base in precision index, Bert-base is superior to the pre-trained language combined neural network in performance. The reason is that fewer training data and sparse text features of short text data make it difficult to fully capture text information. IRDP-STC is better than the baseline model in the four evaluation indexes, and the four evaluation indexes are 1.1 percentage points higher than the optimal baseline model of each index, indicating that IRDP-STC can fully learn the characteristics of short text data and prove the effectiveness of the model.

C. Low-resource scenario experiment

In the actual business scenarios, there may be insufficient training corpus. DPCNN, the best performance baseline model based on neural network, Bert-base, Bert-base-BiLSTM and IRDP-STC in Section 3.2 are selected for low training resource scenario experiments.

The experimental set up is consistent with the previous experiment. Only the training set corpus is modified, and the training set is set to 10%, 20%, 40%, 60%, 80% and 100% of

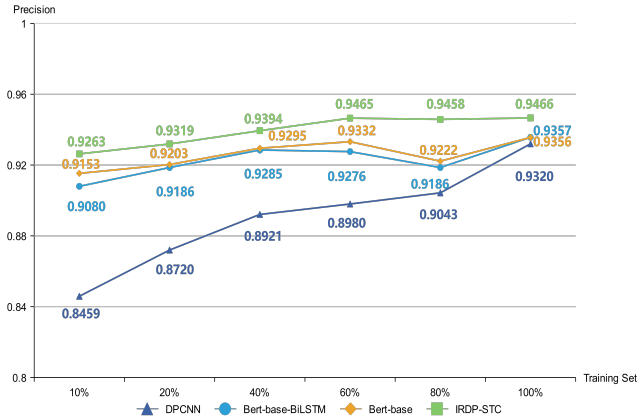

Fig. 5. Precision values of low-resource scenarios for each model.

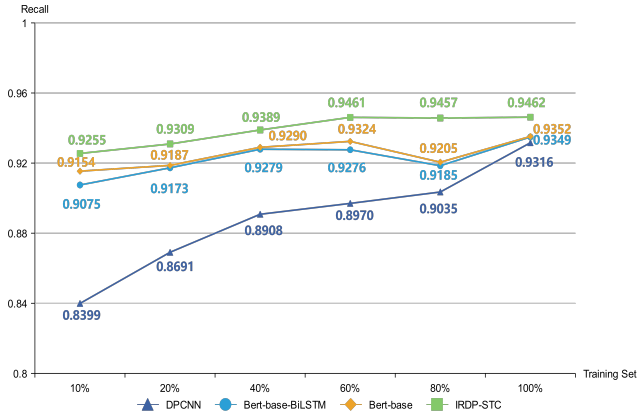

Fig. 6. Recall values of low-resource scenarios for each model.

the original training set corpus respectively. The classification performance of the target model under low resource scenarios is compared. Precision, Recall, F1 and Accuracy of each model under low resource scenarios are shown in Fig.5-8.

It can be seen from the Fig.5-8 that when the training corpus decreases to 10% of the original training corpus, the performance of each model decreases to varying degrees.

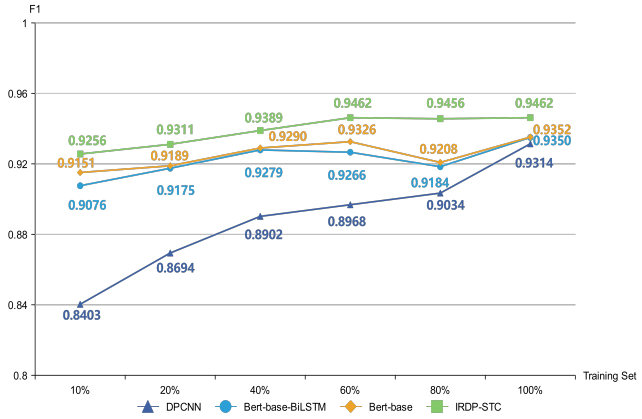

Fig. 7. F1 values of low-resource scenarios for each model.

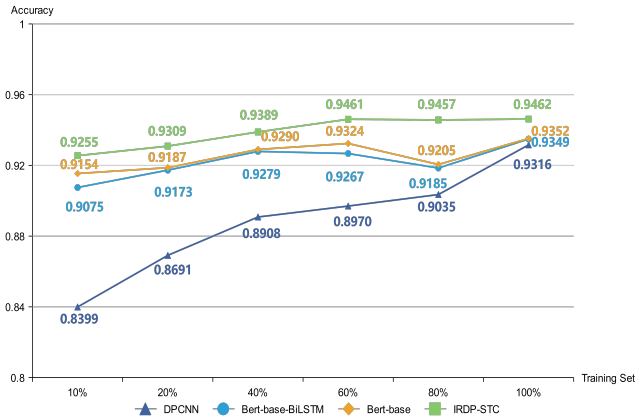

Fig. 8. Accuracy values of low-resource scenarios for each model.

Specifically, the model DPCNN based on neural network has the most dramatic decline, and the four evaluation indexes have an average decline of 9.02%. Bert-base-BiLSTM decreased by 2.75% ; bert-base decreased by 2% ; IRDP-STC decreased by 2.06%. In addition to high performance robustness, IRDP-STC maintains the optimal performance of four indicators in the training corpus of each low resource dimension, which verifies its effectiveness in the low training corpus scenario.

CONCLUSION

This paper proposes a short text classification method IRDP-STC based on R-Drop and pre-trained language model, which makes full use of text features in short-text scenarios and takes into account the model classification performance and model robustness. The effectiveness of the model is verified on THUCNews dataset, and it is proved that the model has better adaptability in low training sample scenario through low resource scenario experiments. At the same time, the method has strong compatibility and can be used to change models and classifiers according to specific data scenarios. For example, in addition to the Bert pre-trained model used in this paper, other pre-trained models such as XLNet can be replaced. However, this paper does not analyze the category imbalance and small sample scenarios, which is also the direction worthy of further research.

REFERENCES

[1] Wu L, Li J, Wang Y, et al, "R-drop: Regularized dropout for neural networks," Advances in Neural Information Processing Systems. vol. 34, pp. 10890-10905, 2021.

[2] Kim Y, "Convolutional neural networks for sentence classification," [Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha Qatar, p. 1746-175, 2014].

[3] Mikolov T, Karafiát M, Burget L, et al, "Recurrent neural network based language model," INTERSPEECH. vol.2(3), pp.1045-1048, 2010.

[4] Graves A, Schmidhuber J, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural networks. vol.18(5-6), pp. 602-610,2005.

[5] Vaswani A, Shazeer N, Parmar N, et al, " Attention is all you need," [Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS), Los Angeles, USA, p. 6000-6010,2017].

[6] Cho K, Van Merriënboer B, Gulcehre C, et al, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," [Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, p. 1724–1734, 2014].

[7] Devlin, J, Chang, M. W, Lee, K, Toutanova, K, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv:1810.04805, 2018.

[8] Lai, S, Xu, L, Liu, K, Zhao, J, "Recurrent convolutional neural networks for text classification," [Twenty-ninth AAAI conference on artificial intelligence, Austin, Texas, USA, 2015].

[9] Johnson R, Zhang T, "Deep pyramid convolutional neural networks for text categorization," [Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, p. 562-570].