Hindawi
Scientific Programming
Volume 2021, Article ID 5753693, 9 pages
https://doi.org/10.1155/2021/5753693

Research Article

Training Method and Device of Chemical Industry Chinese Language Model Based on Knowledge Distillation

Wen-Ting Li, Shang-Bing Gao, Jun-Qiang Zhang, and Shu-Xing Guo

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China

Correspondence should be addressed to Shang-Bing Gao; gaoshangbing@hyit.edu.cn and Jun-Qiang Zhang; zhangjq0906@hyit.edu.cn

Received 16 August 2021; Revised 29 November 2021; Accepted 1 December 2021; Published 13 December 2021

Academic Editor: Wei-Chuen Yau

Recent advances in pretraining language models have obtained state-of-the-art results in various natural language processing tasks. However, these huge pretraining language models are difficult to be used in practical applications, such as mobile devices and embedded devices. Moreover, there is no pretraining language model for the chemical industry. In this work, we propose a method to pretrain a smaller language representation model of the chemical industry domain. First, a huge number of chemical industry texts are used as pretraining corpus, and nontraditional knowledge distillation technology is used to build a simplified model to learn the knowledge in the BERT model. By learning the embedded layer, the middle layer, and the prediction layer at different stages, the simplified model not only learns the probability distribution of the prediction layer but also learns the embedded layer and the middle layer at the same time, to acquire the learning ability of BERT model. Finally, it is applied to the downstream tasks. Experiments show that, compared with the current BERT model distillation method, our method makes full use of the rich feature knowledge in the middle layer of the teacher model while building a student model based on the BiLSTM architecture, which effectively solves the problem that the traditional student model based on the transformer architecture is too large and improves the accuracy of the language model in the chemical domain.

1. Introduction

The past years have seen several major breakthroughs studies that are made in pretrained language models (PLMs) (BERT [1], XLNet [2], RoBERTa [3], SpanBERT [4], and ALBERT [5]). While the learning ability of the current PLMs has been improved a lot significantly, they often have hundreds of millions of parameters, and high computational power should be required. So, it leads to the current PLMS being difficult to apply to solve problems in real life. According to the current research on PLMS, training a large and complex language model still brings great performance on many tasks.

The trend toward bigger models has become inevitable but caused some social concerns. Among them, the most typical one is the BERT model which caused a sensation in the whole NLP world at that time. It had 300 million parameters. The BERT base model is trained on 4 cloud TPUs (16 TPUs in total). BERT large trains on 16 cloud TPUs (64 TPU chips in total). Each pretraining lasts for 4 days. It follows that the training of the BERT has a high requirement for calculation force and memory. While these PLMS applications are used in real-time operations on devices, which may bring better services, the growing computational and memory requirements of these models may hamper wide adoption.

Many researches showed that the domain-specific pretraining language model can perform better in domain tasks. A large number of corpus and reasonable model structures can make the model better improve its learning ability [6]. The chemical industry belongs to the basic economy of our country and is one of the pillar industries of our country. At the same time, the chemical industry plays an important role in China's economic growth. On March 11, 2019, the International Council of Chemical Associations (ICCA) released a report on the analysis of the contribution of the

chemical industry to the global economy. According to the report, the chemical industry is involved in almost all production industries, and its contribution to the global GDP is estimated to be 5.7 trillion US dollars (7% of the global GDP) through direct, indirect, and induced impacts. Therefore, it is very important to create a pretraining language model in the chemical industry, to solve the text problems in the chemical industry more efficiently.

To create a language model in the chemical field, we have to compress the large model and use a large amount of corpus from the chemical field for training. In the compression, we choose the technology of knowledge distillation, which is different from the previous technology of knowledge distillation [7]. It is not only to learn from the probability distribution of the final output of the teacher model but also to learn in the embedded layer, the middle layer, and the prediction layer. Based on the framework, we propose a training method and device of the Chemical Industry Chinese Language Model based on knowledge distillation.

The main contributions of this work are as follows. (1) Traditional knowledge distillation methods on BERT models often failed to fully learn the representational capabilities of each layer of the teacher model, or to learn these, student models based on transformer architecture still needed to be used, and these student models still had a huge number of parameters. Therefore, we proposed a multilayer BiLSTM architecture for student models to fully learn the representational capabilities of the teacher model, which significantly reduced the number of student model parameters at the expense of only a small portion of performance compared to the former. (2) Nowadays, the pretraining language model is more and more huge, which is difficult to apply in real life. To solve this problem, we have constructed a lightweight multilayer BiLSTM architecture for student models to learn the representational capabilities of the teacher model, and our proposed approach combines a certain level of performance with the lightweight, which is more conducive to be applied to real industry-specific tasks. (3) There are currently no pretrained language models specifically applied to specific chemical industry domains. Based on a large chemical industry corpus, we have constructed a framework of distillation pretrained language models specifically for the chemical industry for later application to specific tasks in the chemical industry.

2. Related Work

2.1. Pretrained Language Models (PLMs). Recently, in the field of natural language processing (NLP), the use of language model pretraining has been improved in several NLP tasks and has been widely concerned. The previous researches on language models mainly include feature-based methods and fine-tuning methods [8, 9]. The details are shown in Table 1.

It can be drawn from Table 1 that the feature-based methods are mainly divided into three types. The first type is the context-independent word representation, mainly including word2vec [10], glove [11], and fastText [12]. The second type is sentence-level representation. For example, continuous learning for a sentence using Conceptors was proposed by Liu et al. [13], part-of-speech-based long short-term memory network for learning sentence representations was proposed by Zhu et al. [14], and learning sentence representations from explicit discourse relations was proposed by Nie et al. [15]. The third type is the contextualized word representation; the most typical one is the ELMO model. The main feature of the algorithm is that the representation of each word is a function of the entire input sentence. The specific method is to first train the Bidirectional LSTM model with the language model as the target on the large corpus and then use the LSTM to generate the word representation.

The fine-tuning method is to pretrain the language model on a large corpus without monitoring the target and use the labeled data in the domain to fit the model for subsequent applications. The BERT model is one of the models that use this method, but at the same time, although this training paradigm makes the model perform well, the consequent increase in the number of parameters and the long training time makes the model difficult to be applied to real business scenarios. To solve this problem, the article proposes an effective solution, and this method is also suitable for the recently proposed XLNET, RoBERTa, SpanBERT, ALBERT, and other models.

2.2. Knowledge Distillation. To effectively solve the problem of model oversize, we focus on model compression technology [16–18], which can make the model more concise and conducive to application in real life.

The traditional understanding is that training a deep network requires a large number of connections (weights). However, the network training will lead to a high degree of parameter redundancy. The pruning of the network [19–21], reducing the network connections, is a common strategy for model compression. The other direction is weight quantification. In this case, the connection weights are limited to a set of discrete values, with fewer bits representing the weights. However, most of these pruning and quantification techniques [22, 23] are performed on convolutional networks. Only some jobs are designed for specific structural information (such as deep language models [9, 24–26]).

The goal of knowledge distillation is to compress a network with a large number of parameters into a compact and fast working model. This can be achieved by training the compact model to simulate soft inference to a larger model. Mirzadeh et al. [27] proposed a framework based on teacher assistant knowledge distillation. Liu et al. [28] proposed distilling structured knowledge from large networks to compact networks. For the BERT model of distillation compression, Sun et al. [29] first proposed a framework for distillation of the intermediate layers of the BERT model, which takes full advantage of the rich information in the middle hidden layers of the teacher model and encourages the student model to learn and imitate from the teacher model through multilayer distillation. Jiao et al. [30] proposed a two-stage BERT knowledge distillation learning framework that allowed the use of the above distillation in both pretraining and fine-tuning stages, resulting in richer

Table 1: The description of previous research on language models.

Feature-based methods	Fine-tuning methods
Context-independent word representation Sentence-level representation Contextualized word representation	Pretraining a language model on a large corpus with an unsupervised objective and then fine-tuning the model with in-domain labeled data

knowledge learning. Xu et al. [31] proposed a model compression approach that gradually uses small modules to replace modules in BERT, in which only a loss function and a hyperparameter are used to be able to perform model compression without using transformer specific features for compression, which is a general practice. Fu et al. [32] introduced contrastive learning into the construction of distillation loss functions, and the model performance was improved. Feng et al. [33] solved the problem of poor distillation due to lack of data during distillation by means of cross-domain data enhancement. Chen et al. [34] proposed an extraction-then-distillation strategy that reuses the parameters of the teacher model. This strategy can be used for student models of any size, making the student model primed with certain knowledge before the distillation process begins, speeding up convergence, and improving task agnostic distillation efficiency. We studied the problem of compression of linguistic models on a large scale and proposed a training method and device of the Chemical Industry Chinese Language Model based on knowledge distillation to effectively transfer the knowledge of the teacher to the model of the student.

3. Method

In this section, we propose a training method and device of the Chemical Industry Chinese Language Model based on knowledge distillation. The algorithmic procedure flow chart of the proposed framework is shown in Figure 1.

The proposed algorithmic framework consists of a teacher model, a student model, and, most critically, a loss function to connect the two models. First, the framework trained the teacher model, and the raw corpus text was input to the teacher model for training to obtain the trained teacher model weights. Then, the framework started to perform knowledge distillation by distilling the word embedding layer loss, the intermediate layer loss, and the prediction layer loss of the teacher model, respectively, so that the three layers of loss are distilled into the corresponding three-layer BiLSTM model of the student model. Finally, the knowledge distillation ends, and a student model that has learned the performance of the teacher model is obtained. The detailed program algorithm is shown in Algorithm 1.

3.1. Teacher Model. In the teacher model BERT, the original text corpus set T after special processing is first to read and store in T' after processing by line segmentation. The specific storage format is $T' = \{d_0, d_1, ..., d_i, ...\}$, where d_i is the *i-th* article. d_i stores the collection of all the sentences from article *i*. $d_i = \{l_0, l_1, ..., l_j, ...\}$, where l_j is the *j-th* sentence in d_i, and $l_j = \{t_0, t_1, ..., t_k, ...\}$, where t_k is the *k-th* token in l_j. Next, the order of articles was scrambled, dupe_factor = 10, and then, a random mask was carried out, and 10*len (d_i) bar samples were generated for each article. While the sampled sentence length exceeds the set maximum sentence length Lmax value, the next sentence prediction task in BERT is deleted from the beginning or at the end of one long sentence at random.

Each token in each sentence in T' was sent into BERT's token Embedding Segment Embeddings and Position Embeddings, respectively, and the vector encoding V_1, the sentence encoding V_2, and the Position encoding V_3 were obtained, respectively. The vector V_B is obtained by adding the output of the same three dimensions.

BERT in 12 layers of the transformer is cut into 6 layers of the transformer and then will get V_B that is input into the double transformer; BERT and BERT to teacher model covered the token of the probability distribution of m_t and really covered the token vector said m_s loss calculated according to the following formula, where L_t is the random mask task loss function, and then, we carry on the gradient descent optimization model for teachers.

$$L_t\left(m^s, m^t\right) = -\text{soft max}\left(m^t\right) \cdot \log \_\text{soft max}\left(m^s\right). \quad (1)$$

3.2. Student Model. In the multilayer neural network model of the student model, first, T is the original text corpus set in the pretreatment of the same as the model of teachers and embedding operation, but the word vector dimensions are half the word vector dimension BERT model, the text of the pretreatment and data input to the multilayer neural network model, the model for the length of the three layers of two-way memory network, in the process of training the student model, and student model by studying the teacher model embedded in the layer, hidden layer, and middle prediction for correction of the model. The network structure of the student model is shown in Figure 2.

In the embedded layer, the specific formula for the loss calculation of the vector output of the embedded layer of BERT and the multilayer neural network of the teacher model and the student model is as follows:

$$L_{\text{emb}}\left(s_e, t_e\right) = \text{MSE}\left(s_e W_e, t_e\right), \quad (2)$$

where MSE is the Mean Square Error, and matrix $s_e \in \mathbb{R}^{l \times d'}$ and $t_e \in \mathbb{R}^{l \times d}$ embedded, respectively, the student model and

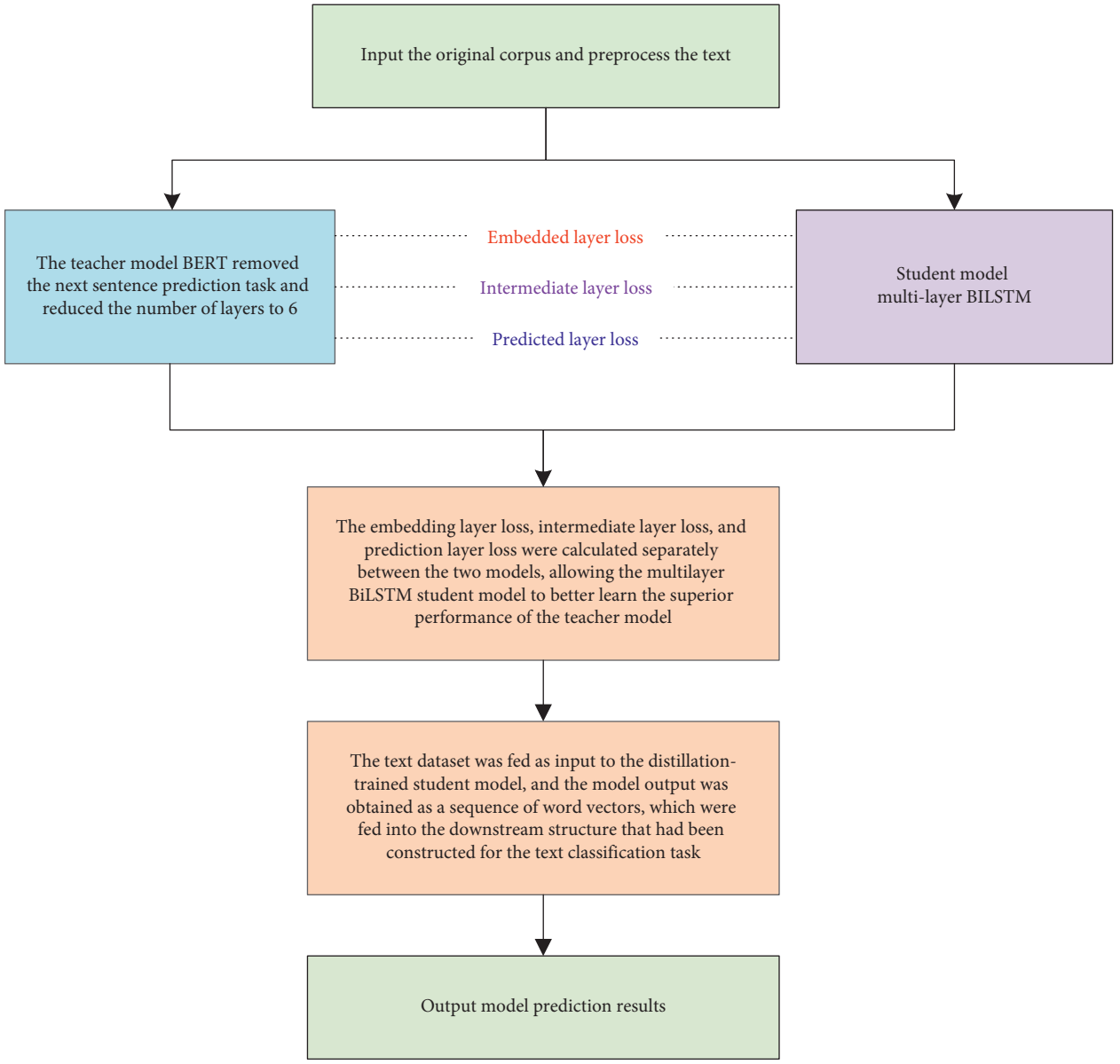

Figure 1: Framework program algorithm flow chart.

Input: Training data x, the trained teacher model Bert and its corresponding label
#Initialization
Randomly initialize student parameters θ
#Starting model distillation
While not converge do
  For batch data set of x
    Extracting word embedding layer loss, intermediate layer loss, and prediction layer loss **from** the teacher model $t_e, t_h, t_p = \text{Bert}(x)$
    Student models begin to distill to learn teacher model knowledge
    Minimizing loss function L_{total}
    Update parameters θ according to gradients
  End for
End while

$$L_{\text{emb}}(s_e, t_e) = \text{MSE}(s_e W_e, t_e),$$
$$L_{\text{hid}}(s_h, t_h) = \text{MSE}(s_h W_h, t_{h'}),$$
$$L_{\text{pre}}(s_p, t_p) = -\text{softmax}(t_p) \cdot \log\_\text{softmax}(s_p/\text{Tem}),$$
$$L_{\text{total}} = \lambda_e L_{\text{emb}}(_e, t_e) + \lambda_{\text{hid}} L_{\text{ht}} + \lambda_{\text{pre}} L_{\text{pre}}(s_p, t_p),$$
$$L_{ht} = \sum_{h=1}^{h=3} L_{\text{hid}}(s_h, t_{2h-1}),$$

Algorithm 1: Continued.

# Input x into the student model BiLSTM to obtain the output as a sequence of word vectors and feed it into the constructed network for the downstream text classification task
While not converge do
  For batch data set of x
    $y = \text{Model}_{\text{classification}}(x)$,
    Returns the maximum value in the corresponding dimension, which is the prediction of the classification result $y_{\text{pred}} = \arg\max(y)$
  End for
End while
Output: prediction results
Remark: θ is hyperparameters, h represents the layer in the student model

Algorithm 1: Distillation to train student models and application of trained student models to downstream text classification tasks.

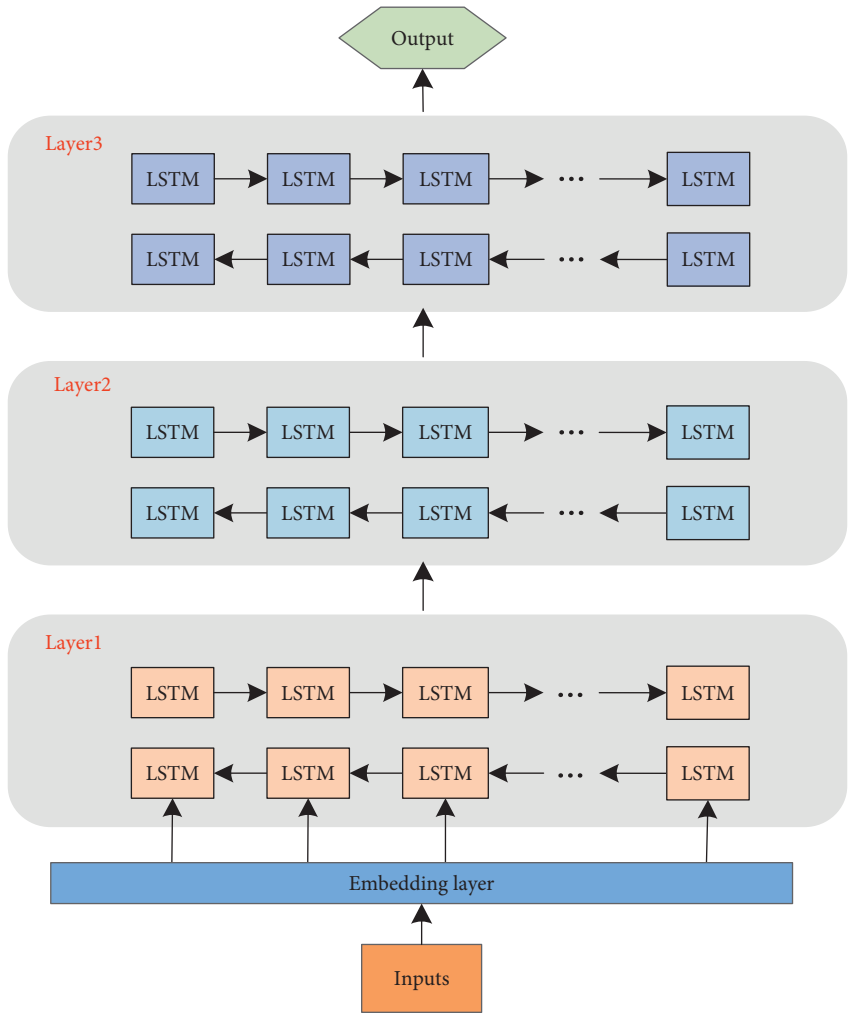

Figure 2: Network structure diagram of the student model.

the teacher said. $L = 128$ represents the text length entered by the model, $d = 768$ represents the hidden layer size of the teacher model, and $d' = 200$ represents the hidden layer size of the student model. In the invention, they have the same shape as the hidden state matrix. Matrix $W_e \in \mathbb{R}^{d' \times d}$ is a study of a linear transformation; it will be the student model said embedded into the same teacher's model of state space.

In the middle hidden layer, the output of each hidden layer in the multilayer neural network of the student model and the output of the hidden layer in the transformer corresponding to the teacher model BERT are calculated with MSE mean square error, and the specific formula is shown as follows.

$$L_{\text{hid}}(s_h, t_h) = \text{MSE}(s_h W_h, t_{h'}), \tag{3}$$

where the matrix $s_h \in \mathbb{R}^{l \times d'}$ and $t_{h'} \in \mathbb{R}^{l \times d}$, respectively, are students and teachers' network output of hidden layers; matrix $W_h \in \mathbb{R}^{d' \times d}$ linear transformation is learning; it will be the student model of hidden state to transform the same teacher model state space.

In the prediction layer, the probability distribution of the output of BERT's softmax layer of the teacher model and the probability distribution of the output of the softmax layer of the multilayer neural network of the student network are calculated as cross-entropy.

$$L_{\text{pre}}(s_p, t_p) = -\text{soft max}(t_p) \cdot \log\_\text{soft max}\left(\frac{s_p}{\text{Tem}}\right), \tag{4}$$

where s_p and t_p are, respectively, the logits output predicted by the student model and the teacher model (the input of the upper layer of softmax). log _soft max is a logarithmic likelihood; Tem = 1 is the temperature value.

By using the above three distillation objectives, the distillation losses of the corresponding layers of the teacher model and the student model can be unified.

$$\begin{aligned} L_{\text{total}} &= \lambda_e L_{\text{emb}}(s_e, t_e) + \lambda_{\text{hid}} L_{ht} + \lambda_{\text{pre}} L_{\text{pre}}(s_p, t_p), \\ L_{ht} &= \sum_{h=1}^{h=3} L_{\text{hid}}(s_h, t_{2h-1}), \end{aligned} \tag{5}$$

where L_{ht} represents the loss formula of the total middle hidden layer and s_h and t_{2h-1}, respectively, represent the hidden layer of the h layer of the student model and the output of the second $2h$-1 layer of the corresponding teacher model. $\lambda_e = 1$, $\lambda_{\text{hid}} = 4$, and $\lambda_{\text{pre}} = 3$, respectively, represent the importance of different layers. The specific algorithm structure is shown in Figure 3.

4. Experiments

In this section, we evaluated the performance of this model under the comparison of different experimental models.

The experimental configuration is an AI server configured with 2× Intel Xeon 6148, 512 g memory, 4× 1.9 t SSD hard disk, raid card, 2× ten thousand network card, 8× Tesla V100 card, 2× double port 100 Gbps HCA card, 3000 W 1 + 1 redundant server power. And the framework selected for the experiment was Tensorflow1.12.0. The data set of the Chinese chemical industry in this experiment is from recruitment information of recruitment websites such as Yingcai, the original corpus dataset with a data size of 1,976,522.

4.1. Model Setup. Due to limited equipment, we only retain transformers with 6 layers in the BERT base model, and according to previous studies, we also abandon the next_-sentence task to carry out implementation research. The research data shows that the BERT model with 6 layers still has good performance.

We created a multilayer BiLSTM neural network with a hidden size of 200 as a student model. For the BERT model after deletion as a teacher model, with the number of layers being 6, the size of hidden layers is 768 and the head number is 12. The layer mapping function between the teacher model and the student model is $f(h) = 2h\text{-}1$. The student model learns at every two layers in the teacher model.

4.2. Teacher Model. Here, we take the BERT model as our teacher model and do not make any settings on the teacher model. Any large pretraining model based on a transformer can be plugged into this framework.

BERT's model architecture is a multilayer bidirectional transformer encoder. BERT base consists of 6 layers, 12 self-attention heads, and 768-dimensional hidden state representation.

Similarly, for the comparison model settings, we changed the layers of the student model accordingly based on the 6-layer teacher model for fairness.

4.3. Student Model. We compare the following models.

4.3.1. Student Model without Distillation. We consider BiLSTM encoders with word embeddings. The last hidden state of BiLSTM is fed into softmax for classification, and the network parameters are trained by optimizing cross-entropy loss over labeled data. We use a basic tokenizer with this model that lowercases all words and splits by whitespace.

4.3.2. Student Model with Distillation. In this, we distill the aforementioned student with (soft/hard) targets and representations from the teacher. First, we fine-tune the teacher on labeled data and use it to generate the logits and hidden state representations for unlabeled instances. We train the student model end-to-end using cross-entropy loss on labeled instances as well as logit loss and representation loss on the unlabeled data. We test three different learning strategies based on a joint optimization scheme as well as two stagewise ones with gradual unfreezing of the intermediate layers.

To verify the validity of the model proposed in this paper, the improved pretraining model is applied to the text classification task. It is noteworthy that we do not compare this model with the recent MobileBERT [35], since the

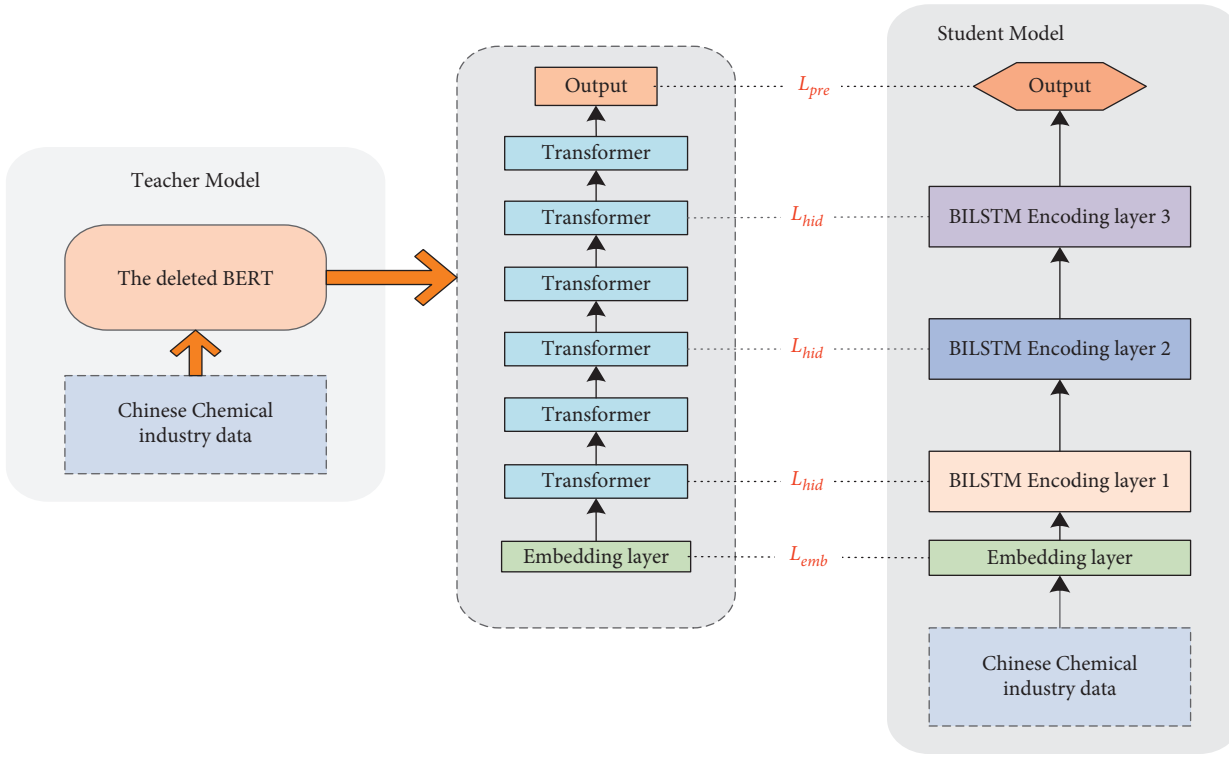

Figure 3: The specific algorithm structure.

MobileBERT model employs the transformer block with different architectures.

We created a multilayer BiLSTM neural network with a hidden size of 200 as a student model. For the BERT model after deletion as a teacher model, with the number of layers being 6, the size of hidden layers is 768 and the head number is 12. The layer mapping function between the teacher model and the student model is $f(h) = 2h\text{-}1$. The student model learns at every two layers in the teacher model. The learning weight of each layer is set to $\lambda_e = 1$, $\lambda_{\text{hid}} = 4$, and $\lambda_{\text{pre}} = 3$, respectively, which performs well for the learning of the student model.

From Table 2, We can find that, compared with the latest method, although our method did not reach the best in the two indicators of accuracy and $F1$ value, it also achieved the third score, which was not a big difference from the latest method, and our proposed method student model was using the BiLSTM architecture with a smaller number of parameters, which made the training speed of the model much faster at the expense of only a small part of the performance. To a certain extent, this also shows that the multilayer BiLSTM architecture can learn the performance of the transformer architecture model very well.

We investigate the effects of distillation objectives on our model learning. Several baselines are proposed including our model learning without the hidden layer distillation (no hidden layer), embedding layer distillation (no embedding layer), and prediction layer distillation (no prediction layer), respectively. The findings are shown in Table 3, which showed that the student model performance could be effectively improved when introducing three layers of loss into the distillation framework, with the model decreasing more significantly when the middle layer of distillation was removed in the ablation experiment, followed by the prediction layer and word embedding layer. To make the student model fully learn the performance of the teacher model, for the intermediate layer which contained the richest knowledge, this method selected a strategy of extracting the intermediate layers of the teacher model at intervals and distilled them to the student model, so that the student model can better characterize the learning teacher

Table 2: Distillation performance with BERT base.

Model	Layers	Hidden	Acc (%)	$F1$ (%)
BERT (teacher)	6	768	94.13	92.52
DistillBILSTM	3	300	91.45	90.21
BERT PKD	3	768	92.87	90.66
DistillBERT [36]	3	768	91.77	89.63
BERT-of-Theseus	3	768	93.43	91.14
BERT-EMD [37]	3	768	93.77	91.34
BiLSTM-KD	3	200	93.13	91.07

Table 3: The experimental situation of the model in the absence of different layers.

Distillation details	Layer	Acc (%)	$F1$ (%)
BiLSTM-KD	All layer	93.13	91.07
	No embedding layer	87.44	85.69
	No hidden layer	74.97	71.57
	No prediction layer	84.22	81.26
BiLSTM	—	72.39	70.06

model performance as a whole. It was also tried to extract only the feature information from the shallow and deeper layers of the teacher model, but the coarse-grained features provided by a single shallow layer or the fine-grained features extracted from the deeper layers could not fully characterize the superior performance of the teacher model, thus resulting in no obvious improvement in the performance of the student model after distillation.

5. Conclusion and Future Work

We proposed a method and device for training Chinese models in the chemical industry based on knowledge distillation. Compared with the traditional distillation model which uses student models based on transformer architecture, this paper constructs a multilayer BiLSTM architecture for student models, so that the superior performance of teacher models can be fully learned using the multilayer structure while further reducing the number of student model participants. Experiments on the text classification task show that the method performed at a somewhat acceptable reduction in performance compared to the baseline model while the number of parameters was significantly reduced, which has important implications for the realistic application of the model to the chemical industry. In future work, we can further consider how to balance the relationship between the number of student model parameters and learning ability, so as to allow the model to be better applied in the industry.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was sponsored by the National Key R&D Program of China (No. 2018YFB1004904), the National Natural Science Foundation of China (61976118), the Key Project of Jiangsu Provincial Department of Education (No. 18KJA520001), Six Talent Peaks Project in Jiangsu Province (XYDXXJS-011), and Jiangsu 333 Engineering Research Funding Project (BRA2016454).

References

[1] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[2] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: generalized autoregressive pretraining for language understanding," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5753–5763, Vancouver, Canada, December 2019.

[3] Y. Liu, M. Ott, N. Goyal et al., "Roberta: a robustly optimized bert pretraining approach," 2019, https://arxiv.org/abs/1907.11692.

[4] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: a lite bert for self-supervised learning of language representations," 2019, https://arxiv.org/abs/1909.11942.

[6] J. Lee, W. Yoon, S. Kim et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[7] D. Araci, "Finbert: financial sentiment analysis with pre-trained language models," 2019, https://arxiv.org/abs/1908.10063.

[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative pre-training," 2018.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[13] T. Liu, L. Ungar, and J. Sedoc, "Continual learning for sentence representations using conceptors," 2019, https://arxiv.org/abs/1904.09187.

[14] W. Zhu, T. Yao, W. Zhang, and B. Wei, "Part-of-speech-based long short-term memory network for learning sentence representations," *IEEE Access*, vol. 7, pp. 51810–51816, 2019.

[15] A. Nie, E. Bennett, and N. Goodman, "DisSent: learning sentence representations from explicit discourse relations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4497–4510, Florence, Italy, July 2019.

[16] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: a survey," 2020, https://arxiv.org/abs/2006.05525.

[17] M. Gupta, V. Varma, S. Damani, and K. N. Narahari, "Compression of deep learning models for NLP," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 3507-3508, Bangalore, India, October 2020.

[18] D. H. Le, V. T. Nhan, and N. Thoai, "Paying more attention to snapshots of iterative pruning: improving model compression via ensemble distillation," 2020, https://arxiv.org/abs/2006.11487.

[19] P. Molchanov, A. Mallya, S. Tyree, Frosio, and Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, Long Beach, CA, USA, June 2019.

[20] X. Dong and Y. Yang, "Network pruning via transformable architecture search," in *Proceedings of the Neural Information*

Processing Systems (NeurIPS), pp. 760–771, Vancouver, Canada, December 2019.

[21] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," 2020, https://arxiv.org/abs/2003.03033.

[22] J. Wang, M. Ramajayam, E. Charrault, and N. Stanford, "Quantification of precipitate hardening of twin nucleation and growth in Mg and Mg-5Zn using micro-pillar compression," *Acta Materialia*, vol. 163, pp. 68–77, 2019.

[23] A. Carpentier, J. Eisert, D. Gross, and R. Nickl, "Uncertainty quantification for matrix compressed sensing and quantum tomography problems," *High Dimensional Probability VIII*, Birkhäuser, Cham, Switzerland, pp. 385–430, 2019.

[24] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," 2019, https://arxiv.org/abs/1905.04226.

[25] X. Cheng, "Dual-view distilled BERT for sentence embedding," 2021, https://arxiv.org/abs/2104.08675.

[26] P. Liu, X. Wang, L. Wang, W. Ye, X. Xi, and S. Zhang, "Distilling knowledge from BERT into simple fully connected neural networks for efficient vertical retrieval," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3965–3975, 2021.

[27] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5191–5198, Vancouver, Canada, February 2020.

[28] Y. Liu, C. Shu, J. Wang, and C. Shen, "Structured knowledge distillation for dense prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020.

[29] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," 2019, https://arxiv.org/abs/1908.09355.

[30] X. Jiao, Y. Yin, L. Shang et al., "TinyBERT: distilling bert for natural language understanding," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 4163–4174, Punta Cana, Dominican Republic, December 2020.

[31] C. Xu, W. Zhou, T. Ge, F Wei, and M Zhou, "Bert-of-theseus: compressing bert by progressive module replacing," 2020, https://arxiv.org/abs/2002.02925.

[32] H. Fu, S. Zhou, Q. Yang et al., "LRC-BERT: latent-representation contrastive knowledge distillation for natural language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, Article ID 12830, Vancouver, Canada, March 2021.

[33] L. Feng, M. Qiu, Y. Li, H. Zheng, and Y. Shen, "Learning to augment for data-scarce domain BERT knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7422–7430, Vancouver, Canada, February 2021.

[34] C. Chen, Y. Yin, L. Shang et al., "Extract then distill: efficient and effective task-agnostic bert distillation," 2021, https://arxiv.org/abs/2104.11928.

[35] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: a compact task-agnostic bert for resource-limited devices," 2020, https://arxiv.org/abs/2004.02984.

[36] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter," 2019, https://arxiv.org/abs/1910.01108.

[37] J. Li, X. Liu, H. Zhao, R. Xu, M. Yang, and Y. Jin, "BERT-EMD: many-to-many layer mapping for bert compression with earth mover's distance," 2020, https://arxiv.org/abs/2010.06133.