MS-BERT: A Multi-layer Self-distillation Approach for BERT Compression Based on Earth Mover's Distance

Jiahui Huang, Bin Cao(✉), Jiaxing Wang, and Jing Fan

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
{huangjh,bincao,wjx,fanjing}@zjut.edu.cn

Abstract. In the past three years, the pre-trained language model is widely used in various natural language processing tasks, which has achieved significant progress. However, the high computational cost has seriously affected the efficiency of the pre-trained language model, which severely impairs the application of the pre-trained language model in resource-limited industries. To improve the efficiency of the model while ensuring the model's accuracy, we propose MS-BERT, a multi-layer self-distillation approach for BERT compression based on Earth Mover's Distance (EMD), which has the following features: (1) MS-BERT allows the lightweight network (student) to learn from all layers of the large model (teacher). In this way, students can learn different levels of knowledge from the teacher, which can enhance students' performance. (2) Earth Mover's Distance (EMD) is introduced to calculate the distance between the teacher layers and the student layers to achieve multi-layer knowledge transfer from teacher to students. (3) Two design strategies of student layers and the top-K uncertainty calculation method are proposed to improve MS-BERT's performance. Extensive experiments conducted on different datasets have proved that our model can be 2 to 12 times faster than BERT under different accuracy losses.

Keywords: Pre-trained language model · BERT · Self-distillation · Multi-layer · EMD

1 Introduction

Nowadays, there are many collaborative management systems, which can realize the assignment and management of tasks by managers. Improving the performance of the system can improve the system's quality of service. For example, the telecommunications complaint system is an agent collaboration system. During the complaint dialogue between the user and the agent, a large number of customer complaints are recorded as the text will be generated. The agent can assign the complaint task to the corresponding business staff for handling. The task allocation process can be regarded as a text classification task in natural

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 407, pp. 316–334, 2021.
https://doi.org/10.1007/978-3-030-92638-0_19

language processing (NLP). With the proposal of Transformer-based language models, such as BERT [1], GPT-2 [17], and XLNet [27], NLP has entered a new era. These transformer-based language models have achieved great success through pre-training on large-scale corpus and fine-tuning on downstream NLP tasks. Applying these models to collaborative systems can greatly improve the performance of the models. However, these transformer-based language models suffer from the following problems: (1) Due to the continuous increase of model parameters, the amount of calculation of inference also increases, which will cause the model's inference speed to be very slow. (2) In the service industry, time and resources are often limited, where makes these language models hard to come into service. Therefore, how to reduce the computational cost and accelerate inference speed has become a widespread concern. Only in this way can the pre-trained language model be better put into use in the industry.

Based on the above questions, many existing studies of Transformer-based language models have tried to accelerate inference speed and reduce the amount of calculation in various aspects, such as weight pruning [2,14], parameter sharing [5,24], low-rank decomposition [10], and knowledge distillation (KD) [6]. Among them, KD is considered the most popular and practical method at present [3]. It is the process of inducing small models (student models) to train through a well-trained large model (teacher model). However, KD also has some problems: (1) KD requires an additional pre-training language model structure, which will put more pressure on deploying these models. (2) The pre-trained language model has been proven to contain a lot of redundant calculations. In industry, the demand for services varies significantly over time. For example, during holidays, the number of complaints may be several times more than on workdays. The pre-trained language model, which after knowledge distillation, cannot cope with the rapid changes in demand due to redundant calculations.

Aiming at the problems of KD, self-distillation can solve them well. Self-distillation [12] does not require additional external resources where the output of teachers and students are in the same model. Moreover, its unique sample-wise adaptive mechanism [12] can well solve the problem of demand changes. Although the previous self-distillation method [12,25] is effective, there are still some limitations: (1) The existing self-distillation method only uses a specific teacher layer to guide the student layers. For example, FastBERT [12] only uses the last teacher layer to guide all student classifiers. However, this method of knowledge transfer is only based on experience without a theoretical basis. The research of Jawahar [7] has proved that different layers of BERT learn different levels of knowledge. The surface information features are in the bottom networks, the syntactic information features are in the middle layer networks, and the semantic information features are in the upper networks. Different NLP tasks require different levels of knowledge contained in different layers of BERT. Therefore, only learning from the specific layer will lead to the lack of partial knowledge and reduce the performance of the models. (2) Previous research directly spliced a student classifier behind each Transformer layer without considering the complexity of different task samples [23]. Some samples with higher complexity may be considered correct in the low-level student classifiers and

output early, which results in a decrease in the models' performance. (3) The information entropy [20] of the complex dataset that contains hundreds of classes is very large, which makes the sample-wise adaptive mechanism [12] useless.

To solve the issues mentioned above, we propose a multi-layer self-distillation BERT based on Earth Mover's Distance (EMD) [16]. First, we design a hierarchical mapping relationship based on EMD where each student classifier can learn knowledge from multiple teacher layers. And the transfer of knowledge is no longer subject to a specific teacher layer. In this way, the students can adaptively learn from various teachers regarding different data sets or NLP tasks. Then, to further improve the performance of our model, we propose two splicing methods of student classifiers, which divide into splicing student classifiers after every k Transformer layer and splicing student classifiers after the last k Transformer layers. It can reduce the error output of samples with higher complexity in the low-level classifiers. Finally, we propose a calculation method of top-K uncertainty, where K refers to the max top K values in the sample output distribution. This calculation method solves the problem that the samples' uncertainty is very high when the sample information is too large and can effectively reduce the samples' uncertainty.

The main contributions of this paper are summarized as follows:

- A multi-layer self-distillation approach is proposed, which can learn the rich linguistic knowledge in the different layers of BERT. In addition, MS-BERT can apply to various BERT-like models, and the best-performing model can be selected for different services to improve the quality of service.
- We use EMD to design a new method of calculating the difference between teachers and students and then find the best way of knowledge transfer.
- We propose two student classifier splicing methods which divide into every k-layers splicing and last k-layers splicing. They can further improve the accuracy of the model. For different data sets, we can choose different splicing strategies.
- We use the top-K strategy to calculate the uncertainty of samples. It can effectively reduce the complexity of the sample so that the model can apply to various data sets.
- We have conducted extensive experiments on public data sets and real data sets. Experiment results prove that our method is effective.

The rest of this paper is organized as follows. In Sect. 2, we will summarize the preliminaries. Section 3 gives a detailed introduction to our method. Section 4 analyzes the validity of the experimental results. In Sect. 5, we will introduce related work of knowledge distillation. Finally, we conclude this paper in Sect. 6.

2 Preliminaries

In this section, we will introduce some preliminary content, which is very important to understand MS-BERT. We first introduce the self-distillation method and then introduce the sample-wise adaptive mechanism.

2.1 Self-distillation

Knowledge distillation refers to transferring the knowledge of the pre-trained teacher model to the student model through distillation. Self-distillation means the transfer of knowledge to oneself.

There are currently two ways of self-distillation: distilling knowledge from the past model to the present (BERT_{SDV}) [25] and distilling knowledge from the high-layer to the low-layer (FastBERT) [12]. Because the self-distillation method in BERT_{SDV} has nothing to do with inference acceleration, and our method aims at inference acceleration. So, we mainly describe the self-distillation in FastBERT. FastBERT is divided into two parts: backbone and branches. The backbone is a well-trained BERT model, and the branches are the student classifiers spliced behind each Transformer layer. The model takes the output of the backbone as a high-quality soft target and extracts it to train the student classifiers. And it uses Kullback-Leibler (KL) divergence to measure the difference between student S and teacher's soft target T as $D_{KL}(S,T)$. The formula is as follows:

$$D_{KL}(S,T) = \sum_{i=1}^{L} S(i) \cdot \log \frac{S(i)}{T(i)} \tag{1}$$

where L is the number of data set categories.

Except for the last layer, there is a student classifier after each Transformer block. FastBERT defines the number of student classifiers as $N-1$, where N is the number of Transformer blocks. It uses the sum of KL divergence to calculate the overall loss of the model as in:

$$Loss = \sum_{i=1}^{N-1} D_{KL}(S_i, T) \tag{2}$$

where S_i refers to the output distribution of the i-th student classifier.

According to the above method, FastBERT can distill knowledge from the high-layer Transformer block to the low-layer student classifiers. Furthermore, it has no additional pre-training structure. The input and output of the teacher and the student are in the same model, so it is called self-distillation.

2.2 Sample-wise Adaptive Inference

The adaptive inference is to control the output depth of samples in the deep network through adaptive calculation [4], and it can even control the complexity of the model. Specifically, given a sample sequence, each level of student classifier will have an output distribution for the sequence. The uncertainty of the sample output distribution is calculated by normalized entropy. The calculation formula is as follows:

$$Uncertainty = \frac{\sum_{i=1}^{L} S(i) \log S(i)}{-\log L} \tag{3}$$

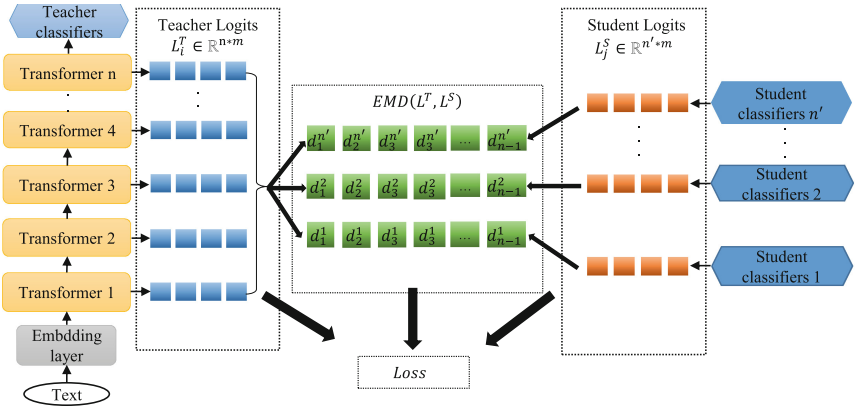

Fig. 1. An ensemble of MS-BERT, which distills knowledge from the N-layer Transformer blocks to each student classifier. The stitching method of the student classifier is stitching every k layer. L^T and L^S are the output distributions of teachers and students, respectively. Calculate the distance between them by EMD and d_i^j represents the output distance between the i-th Transformer layer and the j-th student classifier. n and n' are the number of Transformer layers and student classifiers, respectively. m is the number of categories in the dataset.

where $S(i)$ refers to the value of the i-th label in the sample output distribution. The higher the uncertainty, the greater the amount of information contained in the sample, and the more incorrect the sample.

Subsequently, FastBERT sets a threshold *speed* between 0 and 1 to compare with the uncertainty. Once samples' uncertainty is lower than *speed*, they are considered to be correct enough and will be output in the current student classifier in advance. Otherwise, the samples will go to higher layers for calculation. As the *speed* increases, there will be fewer and fewer samples output in the high-layer classifiers. The comparison method of uncertainty and *speed* avoids that all samples are output in the last layer, and samples can be adaptively output in different layers. It dramatically reduces the inference computation. The above process is the specific process of adaptive inference.

3 Methodology

In this section, we propose a self-distillation BERT based on multi-layer knowledge transfer. In addition, we present two splicing strategies for student layers and a calculation method for top-K uncertainty. Next, we first give an overall overview of our model and approach. Then, we provide a detailed description of the three proposed methods.

3.1 Overview of MS-BERT

The main idea of MS-BERT is to transfer knowledge from different layers of BERT to the student classifier. Like FastBERT [12], our model also contains two parts: backbone and branch, as shown in Fig. 1. The backbone is the BERT-base[1] model officially released by Google. The branch part is mainly some student classifiers spliced behind Transformers, which is mainly used to balance the prediction speed and performance of the model.

Unlike FastBERT, we propose a self-distillation method that utilizes the knowledge contained in all Transformer blocks. Each student classifier in the branch learns from different layers of BERT. In this way, the rich knowledge learned by the 12 layers BERT model can fully transfer to student classifiers, and the performance of all student classifiers can further enhance. In addition, we no longer simply set a classifier after each Transformer, which will cause some high-complexity samples [23] to be output incorrectly in the lower layer. We designed two classifier design strategies to further improve the model's performance by reducing the number of student classifiers. Moreover, we propose a calculation method of top-K uncertainty to reduce the information entropy [20] of the sample. Next, we first introduce the multi-layer self-distillation method based on EMD and then introduce two student classifier splicing strategies. Finally, we will introduce the calculation method of top-K uncertainty.

3.2 Self-distillation with Earth Mover's Distance

BERT encodes a wealth of hierarchical linguistic information. The study of [7] shows that different layers of BERT encode various levels of knowledge. Specifically, the lower layer encodes phrase information and special symbols. In addition, the lower layer still encodes the token's position adequately, while the upper layer has lost the position information. At the same time, the high-layer semantic knowledge will have a feedback effect on the intermediate layer syntactic knowledge, and the intermediate layer syntactic features will be corrected through the high-layer semantic guidance. Therefore, we designed a multi-layer knowledge extraction method based on Earth Mover's Distance (EMD) [16].

EMD is a histogram similarity measure based on the efficiency of the transportation problem, and it is a linear programming problem that has been well solved. It extends the distance between individual elements to the distance between distributions, which can measure the difference between distributions. Our self-distillation method calculates the difference between the output distribution of all Transformer blocks and the student classifiers. Then we use it as the sum of transfer knowledge. Our goal is to reduce this difference to make the student more powerful. The specific process is as follows:

First, in the model training stage, given a text t of length n, we use WordPiece embedding [1] in the embedding layer and encode it as a vector $v = [x_1, x_2, ...x_n]$,

[1] https://github.com/google-research/bert.

where each x_i is constructed by summing the token, segment and position embeddings. We can see how it is calculated as:

$$v = Token\_emd(t) + Seg\_emd(t) + Pos\_emb(t) \tag{4}$$

where $Token\_emd()$ is the token embedding of t, $Seg\_emd()$ is the segment embedding of t, $Pos\_emb()$ is the position embedding of t.

After the embedding is complete, layer-by-layer feature extraction will perform in the multi-layer Transformer blocks, and there will be a probability output in each layer. We use $Logits^T = [Logits_1^T, Logits_2^T, ...Logits_N^T]$ as the output of all Transformer layers, where N represents the number of Transformer, and $Logits_i^T$ represents the output of the i-th layer of Transformer. Similarly, we define the output of each layer of the student classifier as $Logits_j^S$, where j represents the j-th student classifier. We use EMD to calculate the distance between the distribution of students and teachers as:

$$Distance_i^j = Emd\_samples(Logits_i^T, Logits_j^S) \tag{5}$$

where $Distance_i^j$ represents the distance between the output of the i-th layer of Transformer and the output of the j-th layer of student classifier.

Then, we try to get the difference between the j-th student classifier and the overall teacher as $Difference_j$, which makes no knowledge loss in the self-distillation process. Student classifiers can learn different levels of knowledge. The calculation formula is as follows:

$$Difference_j = \sum_{i=1}^{N-1} Distance_i^j \cdot Logits_i^T \tag{6}$$

Next, we use softmax to normalize the overall difference and use it as the teacher's knowledge carrier p_t. We defined it as:

$$p_t = softmax(Difference_j + \lambda Logits_N^T) \tag{7}$$

Similarly, the knowledge carrier p_s of each layer of students is normalized as:

$$p_{s_j} = softmax(Logits_j^S) \tag{8}$$

After that, we use KL divergence to measure the learning goal of each student classifier:

$$Loss_{KL}(p_{s_j}, p_t) = \sum_{i=1}^{M} p_{s_j}(i) \cdot \log \frac{p_{s_j}(i)}{p_t(i)} \tag{9}$$

where M is the number of data set categories.

Finally, we use the sum of the losses of all student classifiers as the overall loss of self-distillation:

$$Total\_loss = \sum_{j=1}^{N'} Loss_{KL}(p_{s_j}, p_t) \tag{10}$$

where N' is the number of student classifiers.

After the above steps, we successfully transferred all the knowledge in Transformers to each layer of the student classifier. Each student classifier can adaptively learn the knowledge contained in different layers of BERT through EMD. Therefore, we have completed the multi-layer mapping strategy. And then, we only need to reduce the *Total_loss* to narrow the difference between teachers and students.

3.3 Student Classifiers Splicing Strategy

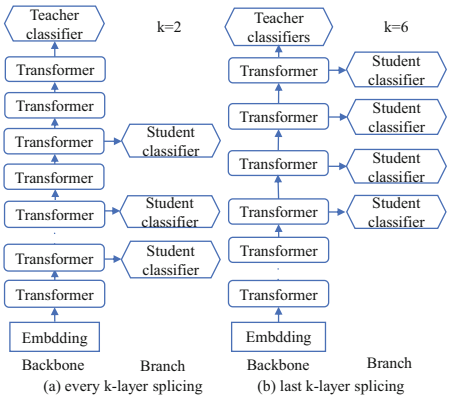

Fig. 2. The model structure of the two splicing strategies. (a) is to splice a classifier every 2 layers; (b) is to splice a classifier in the last 6 layers.

In the process of adaptive inference, the model can exchange for higher efficiency by reducing the accuracy. However, in some experiments, the loss of accuracy is too large. In the service industry, we may not tolerate such a significant loss of accuracy. A reasonable explanation is that different samples have different levels of difficulty [23], and the more difficult samples require a higher-level Transformer to be inferred correctly. The sample-wise adaptive inference mechanism will cause some complex samples to be considered correct in advance and output prematurely, which leads to a decrease in overall performance.

To alleviate this problem, we propose two splicing strategies for student classifiers, as shown in Fig. 2: splicing classifiers every k layers (a); splicing classifiers in the last k layers (b). These two strategies allow the samples to pass through more Transformers before output to determine whether they are correct. In this way, the sample can be more accurate before output. We will verify our strategies in Sect. 4.

3.4 Top-K Uncertainty

For an event or a piece of information, its uncertainty closely relates to the amount of information. For example, we need a lot of information to understand things that are unfamiliar to us. On the contrary, we only need a small amount of information for things we are familiar with. We can use Shannon entropy [20] as a quantitative indicator of information content:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{11}$$

where $p(x_i)$ is the distribution of variable X, and n is the length of the distribution. Uncertainty is the normalized information entropy, as shown in (3). We can calculate the uncertainty of distribution through (3), so as to realize adaptive inference.

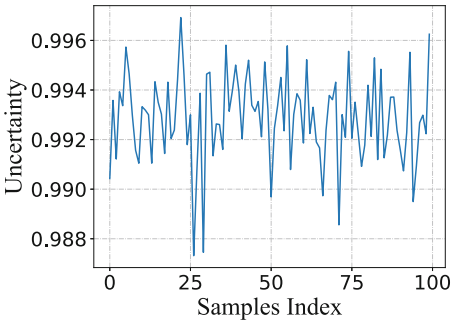

Fig. 3. The uncertainty distribution of the samples. We randomly select 100 texts from the data set and calculate their uncertainty. We can see that the uncertainty of almost all texts is close to 1.

However, complex multi-category data sets, such as the sample of a complaint ticket in the telecommunications industry, contain dozens of categories. Through the calculation of (3), the uncertainty of almost every sample is above 0.99. An intuitive example is shown in Fig. 3. In this way, it is difficult to find a suitable threshold to compare with the uncertainty, which causes the model to lose the ability of adaptive inference. To solve this problem, we propose the calculation method of top-K uncertainty:

$$TopK - uncertainty = \frac{\sum_{i=1}^{K} max_K(p_{s_i} \log p_{s_i})}{-\log K} \tag{12}$$

where K is the top K number in the distribution. In this way, we effectively reduce the uncertainty of the samples, thereby realizing sample-wise adaptive inference for complex data sets. We will verify the effectiveness of our method in Sect. 4.

4 Experiments

In this section, we validate MS-BERT on 4 NLP datasets collected from real-world environments. And we compare MS-BERT with several well-known baselines. A brief introduction of the baselines is as follows.

- **BERT** [1]**.** We use the BERT-base version, which is pre-trained on a large-scale general corpus.
- **BERT-EMD** [11]**.** A knowledge distillation method based on multi-layer knowledge transfer surpasses most of the current knowledge distillation methods in accuracy.
- **FastBERT** [12]**.** The first distillation model combines self-distillation and adaptive inference in NLP tasks.

All experiments are done on a computer with Intel Core(TM) i9-9940X 3.30 GHz CPU and 4 RTX 2080Ti graphics. In the following content, we will first introduce data sets, measurement standards. Then, we compared and analyzed the experimental results. Finally, we conduct an ablation study to analyze the effectiveness of MS-BERT.

4.1 Datasets

We conducted a lot of experiments on public data sets and real-world datasets. These datasets contain three text classification tasks: sentiment analysis, question matching, and complaint service text classification.

(1) *Sentiment analysis.* For the sentiment analysis task, we chose two different datasets. The first one is ChnSentiCorp, a two-category Chinese hotel review dataset with 12,000 texts. The second is the book review [22] dataset, which contains more than 40,000 texts.
(2) *Question matching.* Question matching is a basic task of question answering technology, which is usually regarded as a semantic matching task, and sometimes a paraphrase recognition task. This task aims to search for questions with similar intent to the input questions in the existing database. LCQMC is a large-scale Chinese question matching corpus [13], which contains more than 260,000 question pairs manually labeled and divided into the training set, validation set, and test set.
(3) *Complaint classification.* Telecom complaints service text (TCST) dataset is a real-world complaint text collected from China Telecom. It contains information about users' complaints about telecommunications services. This dataset has a total of 580 categories and a total of 500,000 texts. However, due to data imbalance, the number of texts in many categories is too small, which greatly impacts the overall experimental results. Therefore, we choose 166 categories with a large number of texts for the experiment. We divide the data set into three parts. The training set contains about 280,000 pieces of data, and the validation set and test set each have about 90,000 pieces of data.

To make the experiment fair and reduce the search space of hyperparameters, except for the TCST dataset, the max length of the input sequence of all experiments is 128, the learning rate is 2e−5, and the batch size is 16. For the TCST dataset, the max length of the input sequence is 256, the learning rate is 2e−5, and the batch size is 16. Because there are more than 50% of the texts of telecommunications complaints are longer than 128. Next, we fine-tune all models with three epochs and save the results. Then, we use five epochs for self-distillation.

4.2 Evaluation Metrics

The experiments in this paper mainly have two evaluation indicators: accuracy [19] and Floating-point operations (FLOPs) [12]. Next, we will make a detailed introduction.

(1) **Accuracy.** Accuracy is an evaluation of the overall model, which is widely used in the fields of information retrieval and statistical classification. It can intuitively evaluate the quality of a model. For the binary datasets, the calculation method of accuracy is shown as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{13}$$

where TP refers to the number of positive samples predicted to be positive samples. FP refers to the number of positive samples predicted to be negative samples. TN refers to the number of negative samples predicted to be negative samples. FN refers to the number of negative samples predicted to be positive samples. For a multi-class dataset, the calculation method of accuracy is shown as follows:

$$Accuracy = \frac{\sum TP}{TP + TN + FN + FP} \tag{14}$$

When calculating one of the classes at this time, we treat all other classes as negative classes.

(2) **FLOPs.** FLOPs can use to measure the complexity of an algorithm or model. It represents the amount of model calculation. The smaller the FLOPs of the model, the smaller the amount of calculation required for the model, and the faster the model speed.

4.3 Experimental Result

In this section, we conducted three types of experiments. First, we analyzed the overall performance of the model compared with the baselines. Secondly, we analyzed the effectiveness of the two splicing strategies proposed in this paper. Finally, we analyzed the K value in the uncertainty calculation method. In the following parts, we will analyze the three experiments in detail.

4.3.1 Overall Performance Study

We summarize the overall experimental results on the 4 datasets in Table 1. Compared with BERT-base, MS-BERT can speed up 2 to 12 times under different accuracy losses. We can adjust the speed of the model according to the tolerance for accuracy loss. Compared with BERT-EMD, MS-BERT is slightly insufficient in accuracy. However, MS-BERT far surpasses BERT-EMD in speed when the accuracy is not much different. Moreover, due to the fixed model structure of BERT-EMD, there are a lot of redundant calculations. Redundant calculations make it unable to respond to changes in industrial demand. On the contrary, MS-BERT can effectively reduce redundant calculations due to its adjustable speed. Therefore, our model is more attractive in the industry. Compared with FastBERT, MS-BERT has improved accuracy on all datasets except the TCST dataset. Especially when the *speed* = 0.8, the accuracy rate increases significantly. It increased by 1.16% on ChnSentiCorp, 0.61% on Book review, and 1.57% on LCQMC. Except for the Book review dataset, compared to the increase in accuracy, the increase in FLOPs is not apparent. The reason is that through the multi-layered self-distillation method, the knowledge obtained by each student classifier from the backbone network is more accurate and complete, which makes the output distribution of the student classifier for each sample more accurate. At the same time, the splicing strategy we proposed reduces the error output of samples.

Table 1. Experimental results

Dataset	ChnSentiCorp 12k		Book review 40k		LCQMC 260k		TCST 460k	
Model	*Acc*	*FLOPs* speedup	*Acc*	*FLOPs* speedup	*Acc*	*FLOPs* speedup	*Acc*	*FLOPs* speedup
BERT	94.75	10892M 1.00×	88.38	10892M 1.00×	87.19	10892M 1.00×	57.51	21785M 1.00×
BERT-EMD	92.50	5436M 2.00×	86.11	5436M 2.00×	85.71	5436M 2.00×	55.13	10872M 2.00×
FastBERT (speed = 0.5)	91.25	1696M 6.42×	87.92	3270M 3.33×	84.81	3374M 3.22×	–	–
MS-BERT* (speed = 0.5)	91.83	1817M 6.00×	88.13	3793M 2.87×	85.38	3459M 3.15×	54.93	12151M 1.79×
FastBERT (speed = 0.8)	88.92	1153M 9.44×	85.80	1726M 6.3×	78.98	1736M 6.27×	–	–
MS-BERT (speed = 0.8)	90.08	1204M 9.04×	86.41	1937M 5.63×	80.55	1784M 6.11×	50.48	1854M 11.75×

* For TCST dataset, K is set to 18.

* The student classifiers adopt the way of splicing every two layers of Transformer.

With regard to the Book review dataset, regardless of the speed, the model's accuracy is very close in FastBERT. It shows that the samples' complexity of this dataset is low, and most can predict correctly in the low-level networks. Therefore, as the speed increases, the model's accuracy does not change much, but the acceleration continues to increase. For MS-BERT, due to its low complexity, the student classifier splicing strategy will increase some redundant calculations, which leads to an increase in FLOPs. However, MS-BERT can increase the speed by 2.8 times with almost no loss of accuracy when the *speed* is equal to 0.5. Because the knowledge transfer method of multi-layer self-distillation makes the student classifier's classification performance more robust. Therefore, when the *speed* is not high, the model accuracy loss is small, which is very attractive.

For the TCST dataset, FastBERT loses the ability of adaptive inference because the uncertainty of each sample is too high (see in Fig. 3). Therefore, it cannot achieve inference acceleration, and its experimental results are equivalent to BERT-base. We use '–' to indicate the result. However, based on our method, it can speed up 2 to 12 times under different accuracy losses compared with BERT because the calculation method of top-K uncertainty can greatly reduce the uncertainty of the sample.

4.3.2 Analysis of Different Splicing Strategies

We further studied the performance of two different student classifier splicing strategies. Table 2 summarizes the accuracy and FLOPs of the two strategies under the same speed constraint (*speed* = 0.8). We researched two data sets (ChnSentiCorp and LCQMC). The accuracy of the two strategies improves to different degrees compared with FastBERT. However, the last k-layers splicing strategy will significantly increase the calculation amount of the model, while the increase in the calculation amount of every k-layers splicing strategy can be ignored. A reasonable explanation is that the last k-layer splicing strategy completely eliminates the possibility of samples being output in the lower layers so that samples classified correctly in the lower layers can only be output in the upper layers. The strategy of splicing every k-layer is to prevent samples from being output in each layer so that samples that were initially classified incorrectly can obtain higher correctness, and samples that were initially correct can be output in lower layers.

Table 2. Experimental results of different splicing strategies

Dataset	ChnSentiCorp		LCQMC	
Strategy	*Accuracy*	*FLOPs* speedup	*Accuracy*	*FLOPs* speedup
Every k-layer splicing	90.08	1204M **9.04**×	80.55	1784M **6.11**×
Last k-layer splicing	**93.83**	5688M 1.9×	**87.20**	5803M 1.87×

In summary, we can choose different splicing strategies according to the actual situation of the industry. If the accuracy is more critical, we can choose the stitching strategy of the last k layers. Otherwise, we can choose the stitching strategy of every k layer.

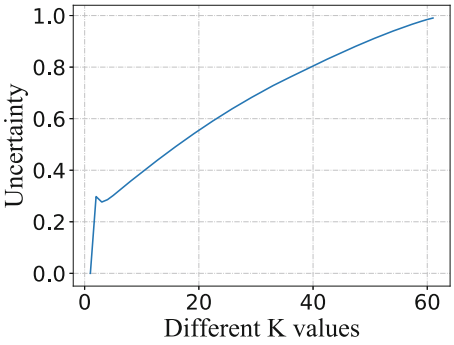

Fig. 4. The relationship curve between K and uncertainty in the TCST dataset.

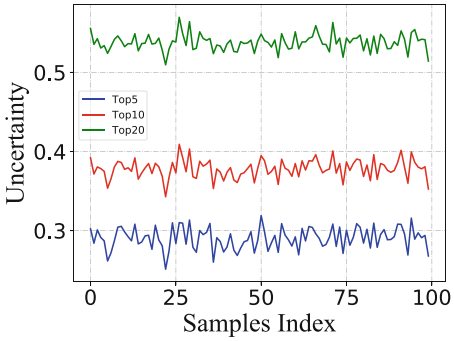

Fig. 5. The uncertainty of different texts in the TCST data set when K is different. We randomly selected 100 pieces of text and set K values to 5, 10, and 20 respectively.

4.3.3 K Value Analysis

For different K values, the uncertainty of the samples is different. We randomly select a sample and calculate its uncertainty as K increases. It can be seen from Fig. 4 that the larger the K values, the higher the uncertainty of the sample. In order to avoid the contingency of a sample, we randomly selected 100 samples and calculated their uncertainty under different K values (5, 10, and 20). Figure 5 shows three different uncertainty curves. It can be proved that when the value of K decreases, the uncertainty decreases. When the data set has a particularly large number of categories, we can realize the adaptive inference of the samples by choosing the appropriate K value. Otherwise, we can make K equal to the number of categories in the data set.

4.4 Ablation Study

To verify the effectiveness of the multi-layer self-distillation method and the student classifier splicing strategy, we conducted ablation experiments on the ChnSentiCorp dataset and the LCQMC dataset. The experimental results are summarized in Table 3, where "w/o multi-layer self-distillation" means only using the last layer of the BERT for self-distillation, and "w/o splicing strategy" refers to splicing student classifiers in each layer of the Transformer.

Table 3. Results of ablation studies

Dataset	*Speed*	ChnSentiCorp		LCQMC	
Method		*Accuracy*	*FLOPs* speedup	*Accuracy*	*FLOPs* speedup
MS-BERT	0.4	92.83	2139M 5.09×	86.14	4051M 2.68×
	0.7	90.67	1446M 7.53×	82.60	2209M 4.93×
W/o multi-layer	0.4	93.58	2302M 4.73×	86.46	4437M 2.45×
Self-distillation	0.7	90.75	1488M 7.3×	83.22	2608M 4.17×
W/o	0.4	92.58	1861M 5.85×	85.76	3658M 2.98M
Splicing strategy	0.7	90.33	1268M 8.58×	81.14	1879M 5.60×

As can be seen from Table 1, MS-BERT has a significant improvement in the model's accuracy. It proves that the multi-layer self-distillation method using EMD and the student classifier splicing strategy is effective. From Table 3, we can find that after removing the multi-layer self-distillation, the model's accuracy rises slightly, and the calculation amount of the model increases significantly. Hence, the overall performance of the model decreases. It shows that it is necessary to use EMD to measure the overall difference between teachers and students, and it plays a key role in reducing the amount of calculation of the model. Because EMD regards the distance between the backbone and the branches as the optimal transmission problem, it learns the optimal multi-layer knowledge transfer method instead of just learning knowledge from a specific layer. The more complete the knowledge learned by the branches, the smaller the amount of calculation they need to process samples. After removing the splicing strategy, the model's accuracy decreases. Because the low-level student classifier has output many errors in advance due to insufficient training depth. Therefore, when adaptive reasoning, we cannot treat all samples equally, but

consider the complexity of the samples. In summary, the two methods we proposed are critical to the improvement of model performance. They can balance the accuracy of the model and the amount of calculation so that the model can be applied to various data sets.

5 Related Work

The related work of our work can be divided into two categories: (1) knowledge distillation, (2) self-distillation.

Knowledge Distillation. There have been many studies on knowledge distillation. The main idea of knowledge distillation is to train a complex teacher network to cultivate a compact student network to have considerable accuracy and efficiency. DistilBERT [18] distilled BERT in the pre-training stage. It used the triple loss to train the student model and achieved great results. TinyBERT [8] proposed a two-stage distillation framework, which performed Transformer distillation during pre-training and fine-tuning. Zhao et al. [28] proposed a new idea to improve the effectiveness of knowledge distillation, bringing down the model's size to several megabytes by reducing the vocabulary. Mirzadeh et al. [15] proved that when the gap between students and teachers is large, the performance of the student network will decrease, so they proposed the concept of assistant teachers. Sun et al. [21] proposed two patient learning strategies so that the student model can learn the intermediate layer of the teacher network. Yang et al. [26] proposed a method for multiple teachers to train a student together so that students can learn knowledge of different tasks. Li et al. [11] proposed a many-to-many mapping for the BERT compression method, namely BERT-EMD, which surpassed most methods in the accuracy of the model.

The general teacher-student framework of knowledge distillation is shown in Fig. 6. The predicted result of the teacher model is divided by the temperature [6], which is regarded as the soft target. The student model obtains knowledge through learning soft targets. However, knowledge distillation requires an additional small model whose performance depends entirely on the teacher model. Deploying additional models in the industry will undoubtedly cause a more significant burden. Moreover, the pre-training language model has proven to contain a lot of redundant calculations [9]. Although knowledge distillation reduces inference calculations, the fixed model structure does not solve the redundancy problem well. Aiming at multi-layer knowledge transfer, unlike BERT-EMD, our method completes knowledge transfer within the same model. And directly use the output distribution of the model to measure the distance. Relatively speaking, the calculation method of BERT-EMD is more complicated.

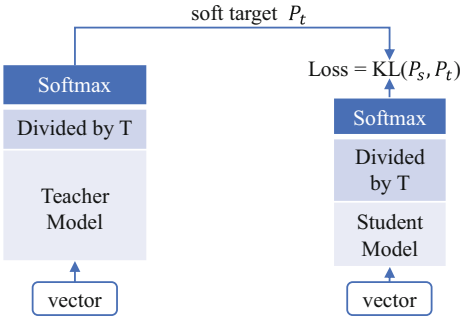

Fig. 6. A general teacher-student framework of knowledge distillation. T stands for temperature, which is used to amplify the knowledge contained in the soft target. The larger the T, the closer the values in the distribution.

Self-distillation. Self-distillation is a unique method of knowledge distillation, whose students and teachers are in the same framework. Liu et al. [12] proposed a FastBERT model, which combined sample adaptive mechanism and self-distillation for the first time. Xu et al. [25] proposed a method that combines self-distillation and self-ensemble to improve further the performance of the model in the fine-tuning stage, but it did not improve the inference speed. We found that the students often transfer knowledge from a specific layer of the teacher model in self-distillation. It will result in ignoring the knowledge contained in other layers of BERT. Because BERT is a multi-layer Transformer structure, different Transformer layers had different natural language features [7].

Different from the self-distillation method, we propose a self-distillation method to learn different levels of knowledge. Our model no longer requires additional external resources and avoids redundant calculations. In addition, it can effectively learn the knowledge contained in different layers of BERT for various NLP tasks.

6 Conclusion

This paper proposes MS-BERT, a multi-layer self-distillation method based on Earth Mover's Distance (EMD). Our model allows each student classifier to learn from all teacher layers in the self-distillation stage, which can reduce the omission of knowledge. Moreover, two student classifier splicing strategies are designed to further enhance the model's performance, thereby reducing the early output of excessive error samples in the adaptive inference stage. In addition, we propose a calculation method of top-K uncertainty to reduce the complexity of the samples. Experimental results conducted on four datasets show that our model has greater accuracy than FastBERT when the inference speed is similar. Compared with the traditional BERT-base model, it can accelerate 2 to 12 times under different accuracy losses. In industry, our model has strong practicality

because of the adjustable inference speed. In the future, we will further study better knowledge transfer methods to improve the model's performance further.

Acknowledgements. This research was partially sponsored by the following funds: National Key R&D Program of China (2018YFB1402800), the Fundamental Research Funds for the Provincial Universities of Zhejiang (RF-A2020007) and Zhejiang Lab (2020AA3AB05).

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics, June 2019
2. Gordon, M., Duh, K., Andrews, N.: Compressing BERT: studying the effects of weight pruning on transfer learning. In: Proceedings of the 5th Workshop on Representation Learning for NLP, pp. 143–155. Association for Computational Linguistics, July 2020
3. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. Int. J. Comput. Vis. **129**, 1789–1819 (2021)
4. Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016)
5. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. CoRR arXiv:1506.02626 (2015)
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
7. Jawahar, G., Sagot, B., Seddah, D.: What does BERT learn about the structure of language? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 3651–3657. Association for Computational Linguistics, July 2019
8. Jiao, X., et al.: TinyBERT: distilling BERT for natural language understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4163–4174. Association for Computational Linguistics, November 2020
9. Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the dark secrets of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 4365–4374. Association for Computational Linguistics, November 2019
10. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. CoRR arXiv:1909.11942 (2019)
11. Li, J., Liu, X., Zhao, H., Xu, R., Yang, M., Jin, Y.: BERT-EMD: many-to-many layer mapping for bert compression with earth mover's distance. arXiv preprint arXiv:2010.06133 (2020)
12. Liu, W., Zhou, P., Wang, Z., Zhao, Z., Deng, H., Ju, Q.: FastBERT: a self-distilling BERT with adaptive inference time. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6035–6044. Association for Computational Linguistics, July 2020

13. Liu, X., et al.: LCQMC: a large-scale Chinese question matching corpus. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, pp. 1952–1962. Association for Computational Linguistics, August 2018
14. McCarley, J., Chakravarti, R., Sil, A.: Structured pruning of a BERT-based question answering model. CoRR arXiv:1910.06360 (2019)
15. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 5191–5198 (2020)
16. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467. IEEE, September 2009
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
18. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR arXiv:1910.01108 (2019)
19. Schütze, H., Manning, C.D., Raghavan, P.: Introduction to Information Retrieval, vol. 39. Cambridge University Press, Cambridge (2008)
20. Shannon, C.E.: A symbolic analysis of relay and switching circuits. Electr. Eng. **57**(12), 713–723 (1938). https://doi.org/10.1109/EE.1938.6431064
21. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 4323–4332. Association for Computational Linguistics, November 2019
22. Torregrossa, F., Claveau, V., Kooli, N., Gravier, G., Allesiardo, R.: On the correlation of word embedding evaluation metrics. In: Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, pp. 4789–4797. European Language Resources Association, May 2020. https://www.aclweb.org/anthology/2020.lrec-1.589
23. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: a multitask benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, pp. 353–355. Association for Computational Linguistics, November 2018
24. Wang, Y., Xu, C., Xu, C., Tao, D.: Packing convolutional neural networks in the frequency domain. IEEE Trans. Pattern Analy. Mach. Intell. **41**(10), 2495–2510 (2018)
25. Xu, Y., Qiu, X., Zhou, L., Huang, X.: Improving BERT fine-tuning via self-ensemble and self-distillation. arXiv preprint arXiv:2002.10345 (2020)
26. Yang, Z., Shou, L., Gong, M., Lin, W., Jiang, D.: Model compression with multi-task knowledge distillation for web-scale question answering system. arXiv preprint arXiv:1904.09636 (2019)
27. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. CoRR arXiv:1906.08237 (2019)
28. Zhao, S., Gupta, R., Song, Y., Zhou, D.: Extreme language model compression with optimal subwords and shared projections. CoRR arXiv:1909.11687 (2019)