Towards Non-task-specific Distillation of BERT via Sentence Representation Approximation

Bowen Wu[1,2], **Huan Zhang**[1]*, **Mengyuan Li**[1]*, **Zongsheng Wang**[2], **Qihang Feng**[2], **Junhong Huang**[2], **Baoxun Wang**[2]
[1]Peking University, Beijing, China
[2]Platform and Content Group, Tencent
jason_wbw,zhanghuan123,limengyuan@pku.edu.cn
jasoawang,careyfeng,vincenthuang,asulewang@tencent.com

Abstract

Recently, BERT has become an essential ingredient of various NLP deep models due to its effectiveness and universal-usability. However, the online deployment of BERT is often blocked by its large-scale parameters and high computational cost. There are plenty of studies showing that the knowledge distillation is efficient in transferring the knowledge from BERT into the model with a smaller size of parameters. Nevertheless, current BERT distillation approaches mainly focus on task-specified distillation, such methodologies lead to the loss of the general semantic knowledge of BERT for universal-usability. In this paper, we propose a sentence representation approximating oriented distillation framework that can distill the pre-trained BERT into a simple LSTM based model without specifying tasks. Consistent with BERT, our distilled model is able to perform transfer learning via fine-tuning to adapt to any sentence-level downstream task. Besides, our model can further cooperate with task-specific distillation procedures. The experimental results on multiple NLP tasks from the GLUE benchmark show that our approach outperforms other task-specific distillation methods or even much larger models, i.e., ELMO, with efficiency well-improved.

1 Introduction

As one of the most important progress in the Natural Language Processing field recently, the Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2019) has been proved to be effective in improving the performances of various NLP tasks by providing a powerful pre-trained language model based on large-scale unlabeled corpora. Recent studies have shown that BERT's capability can be further enhanced by utilizing deeper architectures or performing the pre-training on larger corpora with appropriate guidance (Radford et al., 2019; Yang et al., 2019; Liu et al., 2019b).

Despite its strength in building distributed semantic representations of sentences and supporting various NLP tasks, BERT holds a huge amount of parameters that raises the difficulty of conducting online deployment due to its unsatisfying computational efficiency. To address this issue, various studies have been done to utilize the knowledge distillation (Hinton et al., 2015) for compressing BERT and meanwhile keep its semantic modeling capability as much as possible (Chia et al., 2019; Tsai et al., 2019). The distilling methodologies include simulating BERT with a much smaller model (e.g., LSTM) (Tang et al., 2019b) and reducing some of the components, such as transformers, attentions to obtain the smaller BERT based model (Sun et al., 2019; Barkan et al., 2019).

Nevertheless, the current methods highly rely on a labeled dataset upon a specified task. Firstly, BERT is fine-tuned on the specified task to get the teaching signal for distillation, and the student model with simpler architectures attempts to fit the task-specified fine-tuned BERT afterward. Such methodologies can achieve satisfying results by capturing the task-specified biases (McCallum and Nigam, 1999; Godbole et al., 2018; Min et al., 2019), which are inherited by the tuned BERT (Niven and Kao, 2019; McCoy et al., 2019). Unfortunately, the powerful generalization nature of BERT tends to be lost. Apparently, distilling BERT's original motivation is to obtain a lightweight substitution of BERT for online implementations, and BERT's general semantic knowledge, which plays a significant role in some NLP tasks like sentence similarity quantification, is expected to be maintained accordingly. Meanwhile, for many NLP tasks, manual labeling is quite a high-cost work, and large amounts of annotated data can not be guaranteed to obtain. Thus, it

* Equal contribution during the internship at Tencent.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 70–79

is of great necessity to compress BERT with the non-task-specific training procedure on unlabeled datasets.

For achieving the Non-task-specific Distillation from BERT, this paper proposes a distillation loss function to approximate sentence representations by minimizing the cosine distance between the sentence representation given by the student network and the one from BERT. As a result, a student network with a much smaller scale of parameters is produced. Since the distilling strategy purely focuses on the simulation of sentence embeddings from BERT, which is not directly related to any specific NLP task, the whole training procedure takes only a large amount of sentences without any manual labeling work. Similar to BERT, the smaller student network can also perform transfer learning to any sentence-level downstream tasks, such as text classification and sentence matching. The proposed methodology is evaluated on the open platform of General Language Understanding Evaluation (GLUE) (Wang et al., 2019), including the Single Sentence (SST-2), Similarity and Paraphrase (QQP and MRPC), and Natural Language Inference (MNLI) tasks. The experimental results show that our proposed model outperforms the models distilled from a BERT fine-tuned on a specific task. Moreover, our model inferences more efficiently than other transformer-based distilled models.

2 Related Works

With the propose of ELMo (Peters et al., 2018), various studies take the representation given by pre-trained language models as additional features to improve the performances. Howard and Ruder (2018) propose Universal Language Model Fine-tuning (ULMFiT), an effective transfer learning method that can be applied to any NLP task and accordingly, using pre-trained language models in downstream tasks became one of the most exciting directions. On this basis, developing with deeper network design and more effective training methods, pre-trained models' performances improved continuously (Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019b). Since the release of BERT (Devlin et al., 2019), the state-of-the-art (SOTA) results on 11 NLP tasks have been produced consequently.

With the improvement in performances, the computing cost increases, and the inference procedure becomes slower accordingly. Thus, various studies focused on the model compression upon BERT. Among the most common model compression techniques, the knowledge distillation (Hinton et al., 2015) has been proven to be efficient in transferring the knowledge from large-scaled pre-trained language models into another one (Liu et al., 2019a; Wang et al., 2020; Jiao et al., 2019; Sun et al., 2020). With the help of proposed distillation loss, Sun et al. (2019) compressed BERT into fewer layers by shortening the distance of internal representations between student and teacher BERTs. For the sentence-pair modeling, Barkan et al. (2019) found the cross-attention function across sentences is consuming and tried to remove it with distillation on sentence-pair tasks. Different from these studies distilling BERT into transformer-based models, Chia et al. (2019) proposed convolutional student architecture to distill GPT for efficient text classification. Moreover, focusing on the sequence labeling tasks, (Tsai et al., 2019) derived a BiLSTM or MiniBERT from BERT via standard distillation procedure to simulate the prediction on each token. Besides, Tang et al. (2019a,b) proposed to distill BERT into a BiLSTM based model with penalizing the mean square error between the student's logits and the ones given by BERT as the objective on specific tasks, and introduced various data augmentation methods during distillation.

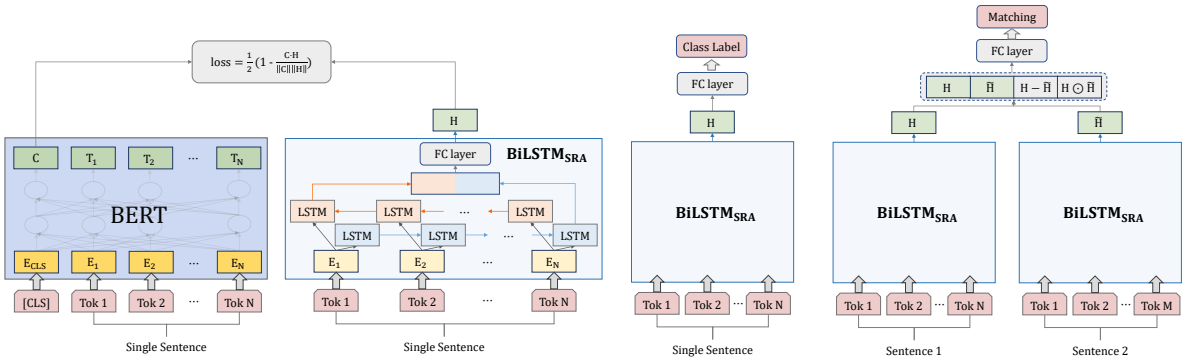
3 Method

As introduced in Section 1, our proposed method consists of two procedures. Firstly, we distill BERT into a smaller student model via approximating the representation of sentences given by BERT. Afterward, similar to BERT, the student model can be fine-tuned on any sentence-level task, such as text classification and sentence matching.

3.1 Distillation Procedure

Suppose $x = \{w_1, w_2, \cdots, w_i, \cdots w_n | i \in [1, n]\}$ stands for a sentence containing n tokens (w_i is the i-th token of x), and let $T : x \to T_x \in \mathbb{R}^d$ be the teacher model which encodes x into d-dimensional sentence embedding T_x, the goal of the sentence approximation oriented distillation is to train a student model $S : x \to S_x \in \mathbb{R}^d$ generating S_x as the approximation of T_x.

In our proposed distillation architecture, as shown in Figure 1a, we take the BERT as the teacher model T, and the hidden representation C is extracted from the top transformer layer upon



(a) Distilling BERT based on Representation Approximation. (b) Tuning (sentence classification). (c) Tuning (sentence-pair oriented).

Figure 1: The illustration of the proposed BERT distillation architecture including the distilling and tuning procedures. Sub-figure (a) demonstrates the distillation procedure taking BERT as the teacher model and BiLSTM as the student model, with the objective of approximating the sentence representations given by BERT. (b) and (c) show two types of fine-tuning frameworks, in which (b) addresses the sentence classification task with the single sentence as the input, and (c) goes for the sentence-pair-oriented tasks, i.e., sentence similarity quantification, natural language inference.

the [CLS][1] token as T_x. For the student model, a standard bidirectional LSTM (BiLSTM) is first employed to encode the sentence into a fixed-size vector H. After that, a fully connected layer without bias terms is built upon the BiLSTM layer to map H into a d-dimensional representation, followed by a $tanh$ activation that normalizes the values of previous representation between -1 and 1 as the final S_x.

As our non-task-specific distillation task has no labeling data, and the signal given by the teacher is a real value vector, it is not feasible to minimize the cross-entropy loss over the soft labels and ground truth labels (Sun et al., 2019; Barkan et al., 2019; Tang et al., 2019b). On this basis, we propose an adjusted cosine similarity between the two real value vectors T_x and S_x to perform the sentence representation approximation. Our distillation objective is computed as follows:

$$\mathcal{L}_{distill} = \frac{1}{2}(1 - \frac{T_x \cdot S_x}{\|T_x\|\|S_x\|}) \quad (1)$$

Here $tanh$ is chosen as the activation function since most values (more than 98% according to our statics) in T_x obtained from BERT are within range of $tanh$ (-1 to 1). The choice of using cosine similarity based loss is mainly based on the following two considerations. Firstly, since 2% values in T_x are outside the range of [-1, 1], it is more reasonable to use a scalable measurement, such as cosine similarity, to deal with these deviations. Secondly, it is meaningful to compute the cosine similarity between sentence embeddings given by BERT (Xiao, 2018).

Overall, after the distillation procedure, we obtained a BiLSTM based "BERT", which is smaller in parameter scale and more efficient in generating a sentence's semantic representation.

Distilling data As our distillation procedure needs no dependency on sentence type or labeling resources but only standard sentences available everywhere, the distillation data selection follows the existing literature on language model pre-training as well as BERT. We use the English Wikipedia to perform the distillation. Furthermore, as the proposed method focus on the sentence representation approximation, the document is segmented into sentences using spacy (Honnibal and Montani, 2017).

3.2 Fine-tuning the Student Model

The fine-tuning on sentence-level tasks is straightforward. The downstream tasks discussed in this paper can be summarized as type judgment on a single sentence and predicting the relationship between two sentences (same as all GLUE tasks). Figure 1b illustrates the model architecture for single sentence classification tasks. The student model S is utilized to provide sentence representation. After that, a multilayer perceptron (MLP) based classifier

[1] [CLS] is a special symbol added in front of other tokens in BERT, and the final hidden state corresponding to this token is usually used as the aggregate sequence representation.

using Relu as activation of hidden layers is applied for the specific task. For the sentence pair tasks, as shown in Figure 1c, the representations H and $\tilde{H}$ for the sentence pair are obtained by transforming two sentences into two BiLSTM based student models with shared weights respectively. Then, following the baseline BiLSTM model reported by GLUE (Wang et al., 2019), we apply a standard concatenate-compare operation between the two sentence embeddings and get an interactive vector as $[H, \tilde{H}, |H - \tilde{H}|, H \odot \tilde{H}]$, where the $\odot$ demotes for the element-wise multiplication. Then, same as the single sentence task, an MLP based classifier is built upon the interactive representation.

For both types of tasks, MLP layers are initialized randomly, and the rest parameters are inherited from the distilled student model. Meanwhile, all parameters are optimized through the training procedure for the specific task.

4 Experimental Setups

4.1 Datasets & Evaluation Tasks

To evaluate the performance of our proposed non-task-specific distilling method, we conduct experiments on three types of sentence-level tasks: sentiment classification (SST-2), similarity (QQP, MRPC), and natural language inference (MNLI). All the tasks come from the GLUE benchmark (Wang et al., 2019).

SST-2 Based on the Stanford Sentiment Treebank dataset (Socher et al., 2013), the SST-2 task is to predict the binary sentiment of a given single sentence. The dataset contains 64k sentences for training and remains 1k for testing.

QQP The Quora Question Pairs[2] dataset consists of pairs of questions, and the corresponding task is to determine whether each pair is semantically equivalent.

MNLI The Multi-Genre Language Inference Corpus (Williams et al., 2018) is a crowdsourced collection of sentence pairs with textual entailment annotations. There are two sections of the test dataset: matched (in-domain, noted as MNLI-m) and mismatched (cross-domain, noted as MNLI-mm).

MRPC The Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) is similar to the QQP dataset. This dataset consists of sentence pairs with binary labels denoting their semantic equivalence.

[2] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

4.2 Model Variations

BERT (Devlin et al., 2019) with two variants: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$, containing 12 and 24 layers of Transformer respectively.

ELMO Baseline (Wang et al., 2019) is a BiLSTM based model, taking ELMo (Peters et al., 2018) embeddings in place of word embeddings.

BERT-PKD (Sun et al., 2019) proposes a patient knowledge distillation approach to compress BERT into a BERT with fewer layers. BERT_3-PKD and BERT_6-PKD stand for the student models consisting of 3 and 6 layers of Transformer, respectively.

DSE (Barkan et al., 2019) is a sentence embedding model based on knowledge distillation from cross-attentive models. For each single sentence modeling, the 24-layers BERT is employed.

$\text{BiLSTM}_{\text{KD}}$ (Tang et al., 2019b) introduces a new distillation objective to distill a BiLSTM based model from BERT for a specific task. $\text{BiLSTM}_{\text{KD}}$+TS (Tang et al., 2019a) donates the distilling procedure performed with the proposed data augmentation strategies.

$\text{BiLSTM}_{\text{SRA}}$ stands for the Sentence Representation Approximation based distillation model proposed in this paper. $\text{BiLSTM}_{\text{SRA + KD}}$ donates performing knowledge distillation method proposed by Tang et al. (2019b) during fine-tuning on a specific task, and $\text{BiLSTM}_{\text{SRA + KD}}$+TS demonstrates using the same augmented dataset to perform the distillation.

4.3 Hyperparameters

For the student model in our proposed distilling method, we employ the 300-dimension GloVe (840B Common Crawl version; Pennington et al., 2014) to initialize the word embeddings. The number of hidden units for the bi-directional LSTM is set to 512, and the size of the task-specific layers is set to 256. All the models are optimized using Adam (Kingma and Ba, 2015). In the distilling procedure, we choose the learning rate as 1×10^{-3} with the batch size=1024. During fine-tuning, the best learning rate on the validation set is picked from $\{2, 3, 5, 10\} \times 10^{-4}$. For the data

#	Models	SST-2	QQP	MNLI-m/mm	MRPC
		Acc	F_1/Acc	Acc	F_1/Acc
1	BiLSTM (report by GLUE)	85.9	61.4 / 81.7	70.3 / 70.8	79.4 / 69.3
2	BiLSTM (report by Tang et al. (2019b))	86.7	63.7 / 86.2	68.7 / 68.3	80.9 / 69.4
3	BiLSTM (our implementation)	84.5	60.3 / 81.6	70.8 / 69.4	80.2 / 69.7
4	ELMO Baseline (Wang et al., 2019)	90.2	65.6 / 85.7	72.9 / 73.4	84.9 / 78.0
5	$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019)	93.5	71.2 / 89.2	84.6 / 83.4	88.9 / 84.8
6	$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2019)	94.9	72.1 / 89.3	86.7 / 85.9	89.3 / 85.4
7	DSE (Barkan et al., 2019)	-	68.5 / 86.9	80.9 / 80.4	86.7 / 80.7
8	BERT_6-PKD (Sun et al., 2019)	92.0	70.7 / 88.9	81.5 / 81.0	85.0 / 79.9
9	BERT_3-PKD (Sun et al., 2019)	87.5	68.1 / 87.8	76.7 / 76.3	80.7 / 72.5
10	$\text{BiLSTM}_{\text{KD}}$ (Tang et al., 2019a)	88.4	- / -	- / -	78.0 / 69.7
11	$\text{BiLSTM}_{\text{SRA}}$ **(Ours)**	90.0	64.4 / 86.2	**72.6 / 72.5**	**83.1 / 75.1**
12	$\text{BiLSTM}_{\text{SRA + KD}}$	**90.2**	**67.7 / 87.8**	72.3 / 72.0	80.2 / 72.8
13	$\text{BiLSTM}_{\text{KD}}$+TS (Tang et al., 2019b)	90.7	68.2 / 88.1	**73.0** / 72.6	82.4 / 76.1
14	$\text{BiLSTM}_{\text{SRA + KD}}$+TS	**91.1**	**68.4 / 88.6**	**73.0 / 72.9**	**83.8 / 76.2**
	Improvements obtained by performing different knowledge distillations				
15	PKD (Sun et al., 2019)	+1.1	+2.3 / +0.9	**+1.9** / +2.0	+0.2 / -0.1
16	KD (Tang et al., 2019a)	+1.7	- / -	- / -	-2.9 / +0.3
17	SRA(**Ours**)	+5.5	+4.1 / +4.6	+1.8 / **+3.1**	**+2.9 / +5.4**
18	SRA(**Ours**)+KD	**+5.7**	**+7.4 / +6.2**	+1.5 / +2.6	0. / +3.1
19	KD+TS (Tang et al., 2019a)	+4.0	+4.5 / +1.9	**+4.3 / +4.2**	+1.5 / **+6.7**
20	SRA(**Ours**)+KD+TS	**+6.6**	**+8.1 / +7.0**	+2.2 / +3.5	**+3.6** / +6.5

Table 1: Evaluation results with scores given by the official evaluation server[3].

augmentation, we use the rule-based method originally suggested by Tang et al. (2019b). Notably, on the SST-2 and MRPC dataset, we stop data augmenting when the transfer set achieves 800K samples following the setting of their follow-up research (Tang et al., 2019a). Besides, inspired by the comparisons in the research of Sun et al. (2019), we find $\text{BERT}_{\text{BASE}}$ can provide more instructive representations than $\text{BERT}_{\text{LARGE}}$. So that, we chose $\text{BERT}_{\text{BASE}}$ as our teacher model to train the non-task-specified $\text{BiLSTM}_{\text{SRA}}$.

5 Results and Analysis

5.1 Model Performance Analysis

For a comprehensive experiment analysis, we collect data and implement comparative experiments on various published BERT and BERT-distillation methods. Table 1 shows the results of our proposed $\text{BiLSTM}_{\text{SRA}}$ and the baselines on the four datasets. All models in the first block (row 1-6) belong to base methods without implementing distillation, the second (row 7-9) and third (row 10-12) blocks show the performances of distillation models using BERT and BiLSTM structures, respectively. Moreover, the fourth block (row 13-14) displays the influences of textual data augmentation approach on our $\text{BiLSTM}_{\text{SRA}}$ and $\text{BiLSTM}_{\text{KD}}$ distillation baseline. The last two blocks contain the results of pure improvements obtained by different distillation methods. To analyze the effectiveness of $\text{BiLSTM}_{\text{SRA}}$ thoroughly, we break down the analyses into the following two perspectives.

5.1.1 Comparison Between Models

Taking those non-distillation methods in the first block as references, $\text{BiLSTM}_{\text{SRA}}$ performs on par with ELMO on all tasks. Especially, $\text{BiLSTM}_{\text{SRA + KD}}$+TS outperforms the ELMO baseline by approximately 3% on QQP and 1% on SST-2 (row 14 vs 4). Such fact shows our compressed "BERT" can provide as good pre-trained representations as ELMO on the sentence-level tasks.

For those distillation methods, both our model and $\text{BiLSTM}_{\text{KD}}$ distill knowledge from BERT into a simple BiLSTM based model, while BERT-PKD focuses on distilling with the BERT of fewer lay-

[3] https://gluebenchmark.com/leaderboard

ers. Despite the powerful BERT based student model and large-scale parameters used by BERT-PKD, our proposed $\text{BiLSTM}_{\text{SRA}}$ still outperforms BERT_3-PKD on SST-2 and MRPC dataset (row 12 vs. 9). For $\text{BiLSTM}_{\text{KD}}$, it proposes a rule-based textual data augmentation approach (noted as TS) to construct transfer sets for the task-specific knowledge distillation. We also employ such method upon $\text{BiLSTM}_{\text{SRA + KD}}$. With and without the data augmentation, $\text{BiLSTM}_{\text{SRA}}$ consistently outperforms $\text{BiLSTM}_{\text{KD}}$ on all tasks (row 12 vs 10; row 14 vs 13). Coworking with the standard knowledge distillation and data augmentation methods, our proposed model is sufficient to distill semantic representation modeled from pre-training tasks as well as the task-specific knowledge included in a fine-tuned BERT.

Besides, DSE's overall architecture is similar to our method for modeling the sentence matching task, except DSE does not reduce the parameter size because it employs the pre-trained $\text{BERT}_{\text{LARGE}}$ to give sentence representations. Thus, on the sentence-pair level tasks, DSE somehow is an upper bound of the distilled models without utilizing any cross attention to model the two sentences' interaction. Comparing with DSE achieved an averaged 80.7 score on all sentence-pair level tasks, $\text{BiLSTM}_{\text{SRA + KD}}$+TS can also obtain 77.2 that only 3.5 points lower (row 7 vs. 14). Analyzing from this fact, our proposed model has distilled a much smaller "BERT" with acceptable performances.

5.1.2 Distillation Effectiveness

Because in each paper, the performances of student models used for distillation vary from each other. To further evaluate the distillation effectiveness, we also report each distillation method's improvement upon the corresponding student directly trained without distillation (in row 15-20). It can be observed that SRA improves the scores by over 3.9% on average, while PKD and KD only provide less than 1.2% increase (row 17/16 vs. 15).

Since our distillation method is unrelated to specific tasks, KD can also be performed upon $\text{BiLSTM}_{\text{SRA}}$ during fine-tuning on a given dataset. This operation provides a notable boost on the QQP task, but damages the performance on both MNLI and MRPC datasets (row 17 vs. 18). We attribute these differences to the following aspects: a) the QQP dataset has more obvious task-specified biases during the sampling process[4]. A pre-trained BERT can not learn such biases; b) a fine-tuned BERT on the MNLI can not further provide more easy-to-use information to guide the student training after performing SRA; c) MRPC does not include enough data to complete KD, which is also indicated by the decreased F1 score shown in row 16 in Table 1. These phenomena reflect that the pre-distillation without paying attention to a specific task can help to learn more useful semantic information from the teacher model.

Different from obtaining the best results on the MNLI dataset, SRA+KD+TS brings few improvements compared to KS+TS (row 19 vs. 20). We attribute this to the difference in the results of pure student BiLSTM between our implementation and the one of Tang et al. (2019b), though our scores are more constant with the baselines given by the GLUE benchmark (Wang et al., 2019).

Models	# of Par.	Inference Time
$\text{BERT}_{\text{LARGE}}$	309 (64x)	1461.9 (54.4x)
$\text{BERT}_{\text{BASE}}$	87 (18x)	479.7 (17.7x)
ELMO	93 (19x)	- (23.7x)
BERT_3-PKD	21 (4x)	- (4.8x)
BERT_6-PKD	42 (9x)	- (9.2x)
DSE	309 (64x)	- (109.1x)
$\text{BiLSTM}_{\text{KD}}$	**2.4 (0.5x)**	31.9 (1.2x)
$\text{BiLSTM}_{\text{SRA}}$	4.8 (1x)	**26.8 (1x)**

Table 2: Comparisons of model size and inference speed. # of Par. denotes the number of millions of parameters, and the inference time is in seconds. The factors inside the brackets are computed comparing to our proposed model.

5.2 Model Efficiency Analysis

To compare the inference speeds of different models, we also implement experiments on 100k samples from the QQP dataset. The results are shown in Table 2. All the inference procedures are performed on a single P40 GPU with a batch size of 1024, respectively. As the inference time is affected by the test machine's computing power, for fair comparisons with ELMO, BERT_3-PKD, BERT_6-PKD, and DSE, we inherit the speed-up factors from previous papers. Besides, the numbers of parameters reported in Table 2 exclude those

[4] https://www.kaggle.com/c/quora-question-pairs/discussion/32819#latest-189493

Models	20%	30%	50%	100%
$\text{BERT}_{\text{LARGE}}$	91.9	92.5	93.5	93.7
BiLSTM	80.7	81.0	83.6	84.5
$\text{BiLSTM}_{\text{KD}}$	81.9	83.2	84.8	86.3
$\text{BiLSTM}_{\text{SRA}}$	**85.9**	**87.3**	**88.1**	**89.2**

Table 3: The accuracy scores evaluated on the SST-2 validation set. The models are trained with different proportions of the training data.

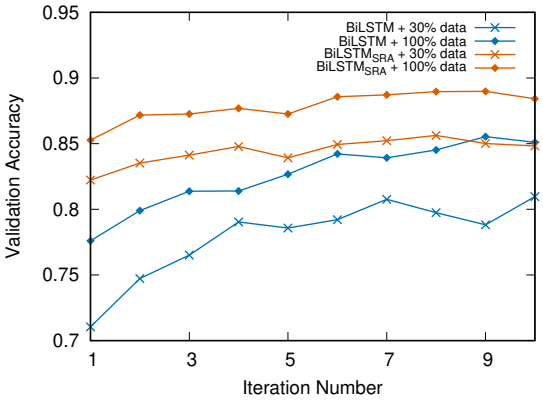


Figure 2: Learning curve on the QQP dataset.

from the embedding layers, since such components do not affect the inference speed and are positively related to the vocabulary sizes, i.e., usually few words appeared for a specific task.

From the results shown in Table 2, it can be observed that the BiLSTM based distilled models have fewer parameters than BERT, ELMO, as well as the other transformer-based models. Compared to the lightest model, both the $\text{BERT}_{\text{BASE}}$ and ELMO are around 20 times larger in parameter size and 20 times slower in inference speed. Even the smallest transformer based model BERT_3-PKD is also four times larger than our proposed $\text{BiLSTM}_{\text{SRA}}$. Comparing with $\text{BiLSTM}_{\text{KD}}$, although our proposed $\text{BiLSTM}_{\text{SRA}}$ is larger in parameter size due to the restriction of the sentence embedding's dimension given by the teacher BERT, it stills inferences more efficiently. This is mainly due to the fact that the more hidden units in $\text{BiLSTM}_{\text{SRA}}$ are more accessible to calculated in parallel by the GPU core, while the larger word embedding size in $\text{BiLSTM}_{\text{KD}}$ slows down its inference efficiency. In conclusion, the cost and production per second of $\text{BiLSTM}_{\text{KD}}$ and $\text{BiLSTM}_{\text{SRA}}$ are within the same scale, but our method achieves better results on GLUE tasks according to the comparison shown in Table 1.

5.3 Influence of Task-specific Data Size

Since pre-trained language models have well-initialized parameters and only learn a few parameters from scratch, these models usually converge faster and are less dependent on large-scale annotations. Correspondingly, the non-task-specific distillation method proposed in this paper also aims to obtain a compressed pre-trained BERT and keep these desirable properties. To evaluate it, in this section, we discuss the influence of the task-specific training data and learning iterations on the performance of our model and the others.

As illustrated in Table 3, we experiment in training the models using different proportions of the dataset. $\text{BERT}_{\text{LARGE}}$ trained on the corresponding data stands for the teacher model of each $\text{BiLSTM}_{\text{KD}}$. No doubt, all the models can achieve better results using more training data, while BERT performs the best. BERT even successfully predicts 91.9% of validation samples under only 20% training data. Comparing with the pure BiLSTM models, the $\text{BiLSTM}_{\text{KD}}$ models slightly improve the performances by 1%$\sim$2%, whereas $\text{BiLSTM}_{\text{SRA}}$ outperforms the best BiLSTM model as well as the $\text{BiLSTM}_{\text{KD}}$ trained with 20% and 30% percent data respectively. Besides, similar to BERT, the difference of accuracy between $\text{BiLSTM}_{\text{SRA}}$ trained with 20% and the one using 100% corpus is relatively small. This phenomenon indicates that our model converges faster and is less dependent on the amount of training data for downstream tasks.

Such conclusions are also reflected in the comparison in Figure 2 of the models' learning curves on QQP. Even though QQP is a large dataset to train a good BiLSTM model, it can be observed that $\text{BiLSTM}_{\text{SRA}}$ trained with 30% data performs equivalent to BiLSTM using the whole corpus. Moreover, using 100% training data, $\text{BiLSTM}_{\text{SRA}}$ even outperforms the converged BiLSTM after the first epoch. Besides, all the $\text{BiLSTM}_{\text{SRA}}$ models converge in much fewer epochs.

5.4 Influence of Distilling Data Size

Despite the task-specified data, Wikipedia corpus is used in the distillation procedure of our proposed method. We also pre-train different $\text{BiLSTM}_{\text{SRA}}$ base models using $\{1, 2, 4\}$ million Wikipedia data, and the corresponding fine-tuning performances on SST-2 and MNLI are reported in Table 4. It can be observed that both the performances of

Size	Distillation	SST-2	MNLI-m
	Loss	Acc	Acc
0M	-	84.5	70.23
1M	0.0288	88.9 (+4.4)	72.01 (+1.78)
2M	0.0257	89.3 (+4.8)	72.09 (+1.86)
4M	***0.0241***	**89.4 (+4.9)**	**72.45 (+2.22)**

Table 4: The distillation losses on the Wikipedia validation set and the accuracy scores of the downstream tasks various with the distillation data sizes.

Models	Acc	F_1
ELMO	65.1	**64.4**
$BERT_{BASE}$ (CLS)	63.9	61.0
$BERT_{BASE}$ (averaged)	**66.4**	64.1
$BiLSTM_{KD}$	56.3	56.6
$BiLSTM_{SRA}$	**62.9**	**61.0**

Table 5: Results of untuned sentence representing models on QQP dataset.

$BiLSTM_{SRA}$ on SST-2 and MNLI are proportional to the distillation loss. This observation indicates the effectiveness of our proposed distillation process and objective.

Besides, distilling with adequate data is sufficient to produce more BERT-like sentence representations as well as achieve better performance on the downstream tasks. Nevertheless, different from the fact that more training data has a significant benefit in a particular task, four times the distilling data can only improve around 0.5 points on both SST-2 and MNLI-m tasks. Thus, our method does not require a vast amount of training data and a long training time to obtain good sentence representations. Furthermore, the second column's loss scores suggest $BiLSTM_{SRA}$ can generate more than 95% similar sentence embeddings with the ones given by BERT under the measure of the cosine similarity.

5.5 Analysis on the Untuned Sentence Representations

A notable characteristic of the pre-trained language models, such as ELMO, BERT, and certainly the non-task oriented distillation models, lies in the capability of providing sentence representations for quantifying similarities of sentences, without any tuning operation based on specific tasks. In this subsection, we conduct the comparisons among models by directly extracting their sentence embeddings without fine-tuning upon sentence similarity oriented tasks.

Table 5 lists the results of models on the QQP dataset. It should be noted that, in this table, ELMO, $BERT_{BASE}$ (CLS) and $BERT_{BASE}$ (averaged) are introduced as the comparison basis, since they can give the SOTA untuned sentence representations for the similarity measurement. The comparison mainly focuses on the performances of our proposed $BiLSTM_{SRA}$ and $BiLSTM_{KD}$. For a thorough comparison, we define the training objective of $BiLSTM_{KD}$ as fitting the cosine similarity score of the sentence pair directly given by the pre-trained $BERT_{BASE}$, which means both the teacher BERT and distilled models do not utilize the labels of QQP dataset. Even though the training goal of $BiLSTM_{KD}$ is more direct than $BiLSTM_{SRA}$, it can be seen that our $BiLSTM_{SRA}$ outperforms the former on the metrics. Furthermore, it achieves scores closed to those of $BERT_{BASE}$. Besides, we can also observe that, for sentence similarity quantification, averaging the context word embeddings as the sentence representation (ELMO and $BERT_{BASE}$ (averaged)) works better than taking the final hidden state corresponding to the [CLS] token ($BERT_{BASE}$ (CLS)).

6 Conclusions

In this paper, we have presented a sentence representation approximating oriented method for distilling the pre-trained BERT model into a much smaller BiLSTM without specifying tasks, so as to inherit the general semantic knowledge of BERT for better generalization and universal-usability. The experiments conducted based on the GLUE benchmark have shown that our proposed non-task-specific distillation methodology can improve the performances on multiple sentence-level downstream tasks. From the experimental results, the following conclusions can be drawn: 1) for a specified task, our proposed distillation method can bring the 5% improvement to the pure BiLSTM model on average; 2) the proposed model can outperform the state-of-the-art BiLSTM based pre-trained language model, which contains much more parameters; 3) compared to the task-specific distillation, our distilled model is less dependent on the corpus size of the downstream task with satisfying performances guaranteed.

References

Oren Barkan, Noam Razin, Itzik Malkiel, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2019. Scalable attentive sentence-pair modeling via distilled sentence embedding. *arXiv preprint arXiv:1908.05161*.

Yew Ken Chia, Sam Witteveen, and Martin Andrews. 2019. Transformer to cnn: Label-scarce distillation for efficient text classification. *arXiv preprint arXiv:1909.03508*.

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

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ameya Godbole, Aman Dalmia, and Sunil Kumar Sahu. 2018. Siamese neural networks with random forest for detecting duplicate question pairs. *arXiv preprint arXiv:1801.07288*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*.

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

Andrew McCallum and Kamal Nigam. 1999. Text classification by bootstrapping with keywords, em and shrinkage. In *Unsupervised Learning in Natural Language Processing*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. *arXiv preprint arXiv:1906.02900*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. 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)*, pages 4314–4323.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,*

ACL 2020, Online, July 5-10, 2020, pages 2158–2170. Association for Computational Linguistics.

Raphael Tang, Yao Lu, and Lin Jimmy. 2019a. Natural language generation for effective knowledge distillation. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019b. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3623–3627.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv: Computation and Language*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.