Multiple Teacher Distillation for Robust and Greener Models

Artur Ilichev[0]
Moscow Institute
of Physics and Technology,
Moscow, Russia
ilichev.as@phystech.edu

Nikita Sorokin **Valentin Malykh** **Irina Piontkovskaya**
Huawei Noah's Ark lab, Moscow, Russia
lastname.firstname@huawei.com

Abstract

The language models nowadays are in the center of natural language processing progress. These models are mostly of significant size. There are successful attempts to reduce them, but at least some of these attempts rely on randomness. We propose a novel distillation procedure leveraging on multiple teachers usage which alleviates random seed dependency and makes the models more robust. We show that this procedure applied to TinyBERT and DistilBERT models improves their worst case results up to 2% while keeping almost the same best-case ones. The latter fact keeps true with a constraint on computational time, which is important to lessen the carbon footprint. In addition, we present the results of an application of the proposed procedure to a computer vision model ResNet, which shows that the statement keeps true in this totally different domain.

1 Introduction

Nowadays the language models became a cornerstone in many natural language processing tasks. Their results in the benchmarks show new high scores. But with great power sometimes comes huge size, the current models could have dozens of billions of weights, e.g. TuringNLG (Rasley et al., 2020), to GPT-3 (Brown et al., 2020) with 175 billion parameters, and counting. In many cases the resources of computational memory are limited and there is a demand for small solutions. One of such solutions is a distillation of language models. There were presented several approaches for the specified task, among others these are TinyBERT (Jiao et al., 2019) and DistilBERT (Sanh et al., 2019).

We analyzed these approaches and found that they share an important flaw - the dependency from the random seed used in the distillation process. During the distillation a student model needs to be trained multiple times with different random seeds to achieve better performance, although it is not guaranteed that there will be "winning numbers" in your seed choice. So we concentrated on the worst case scenario and proposed a technique to improve it. Considering the computational resources, the improvement could be achieved with the same computational budget, allowing one to diminish the carbon footprint. We propose the novel technique of *multi-teacher distillation* called to make the mentioned language models more robust to seed selection. We evaluated our method on a computer vision classification model ResNet (He et al., 2016) and make sure that the proposed technique is applicable to a totally different domain.

Our contribution is as follows: we present (i) a new distillation method and an experimental evaluation of this method for three models, namely (ii) TinyBERT and (iii) DistilBERT, where we modified the distillation procedure adding the task-specific distillation, for three natural language understanding tasks and (iv) ResNet for a computer vision task, showing on the one hand that models learned from multiple teachers are consistently better in the worst case and about the same in the best case, and on the other hand, these models are better with a constraint on computational time.

This work is structured as follows: in Section 3 we describe the distillation process and our modification (in Section 3.5); in Section 4 we describe the datasets used in the experiments, which are described in Section 5. The Section 6 concludes the article and discusses the obtained results.

2 Related Work

Common techniques for model compression and acceleration can be roughly grouped into three groups. **Pruning** parts of large-scale models allows to re-

[0] Work done while Artur Ilichev was at Huawei.

Proceedings of Recent Advances in Natural Language Processing, pages 601–610
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_068

duce the number of weights and accelerate inference. Sajjad et al. (2020) drop entire layers from pre-trained Transformer models, showing that several top-layers can be dropped, maintaining the performance on downstream tasks. Michel et al. (2019) remove all but one attention head, showing that indeed one head might be sufficient at the test time not only for sentence modeling tasks and but for machine translation also. **Quantization** keeps the network structure unchanged, but quantizes network weights to smaller data types, such as int8. Quantization can be performed both post training (Bhandare et al., 2019) or during fine-tuning (Zafrir et al., 2019). **Knowledge distillation** (Hinton et al., 2014) trains the more compact models, students, to reproduce the behavior of a larger model, the teacher. BERT-PKD (Sun et al., 2019), TinyBERT (Jiao et al., 2019) and DistilBERT (Sanh et al., 2019), distilled versions of the BERT model, are commonly used as strong baselines for BERT compression. We describe TinyBERT and DistilBERT models in more detail in the next section. Cho and Hariharan (2019) show that distilling from a better (larger and more accurate) teacher does not always lead to a better student model. We see this result as a motivation for using multiple teachers instead of trying to pick the best one to get a better score.

There are several prior studies considering distillation from **multiple teachers** for Computer Vision (CV) or Natural Language Processing (NLP) tasks. It can be applied to a multi-task or multi-domain setting. For example, Zhang and Peng (2018) combine the knowledge of teachers trained on different tasks, Wu et al. (2019) train teachers on different features extracted from video frames, Ruder et al. (2017) use domain-specific teachers for domain adaptation, and Tan et al. (2019) obtain multilingual machine translation model using teachers pre-trained for each language pair.

Some authors apply multiple teachers without significantly modifying the distillation pipeline, which is closer to our work. Fukuda et al. (2017) propose two ways to utilize multiple teachers in the distillation process: to augment the training data with soft labels provided by different models or to switch the teacher models dynamically at the mini-batch level. Ze et al. (2020) show that averaging the prediction of three teachers trained with different learning rates can improve the score on Question Answering (QA) and Natural Language Inference tasks. Yang et al. (2020) adopted two-stage distillation procedure and showed improvement in several QA tasks. Liu et al. (2019) show the improvement on several tasks from the GLUE benchmark. Additionally, Sau and Balasubramanian (2016) propose to add normally distributed random noise to the logits of the teacher model during distillation, claiming that such procedure is a simulation of learning from multiple teachers.

Besides prediction averaging, multiple teachers can also be utilized to transfer knowledge contained in hidden states or structural relations between examples. You et al. (2017) average soft-labels of multiple teachers and propose to transfer relative dissimilarity among intermediate representations using teacher voting to select the best ordering relationships. Liu et al. (2020) combine soft-labels of multiple teachers with learnable weights, distill structural knowledge between data examples, and transfer intermediate layer representations making each teacher responsible for a specific group of layers in the student network. Both papers relate to the Computer Vision field, both use models with different architectures as teachers, and both show that 5 teachers are better than 3 for their methods (in terms of classification accuracy), but not better for the original knowledge distillation.

To the best of our knowledge, there is no study dedicated to the isolated investigation of the effect that multiple teachers distillation has on the model quality and robustness and of how this effect change with the number of teachers. Importantly, we use models with exactly the same architecture but fine-tuned with different random seeds as teachers.

3 Model Distillation

We briefly describe the formulation of Original Knowledge Distillation procedure (Hinton et al., 2014), two approaches to distill BERT-like models, and one approach to distill the ResNet CV model. Then we describe how multiple teachers get involved in the process.

3.1 Knowledge Distillation

The Original Knowledge Distillation (OKD), proposed by Hinton and co-authors in (Hinton et al., 2014), became an integral part of transferring knowledge from large neural networks to smaller ones. The idea is to train a network called "student" using the task-specific outputs of the so-

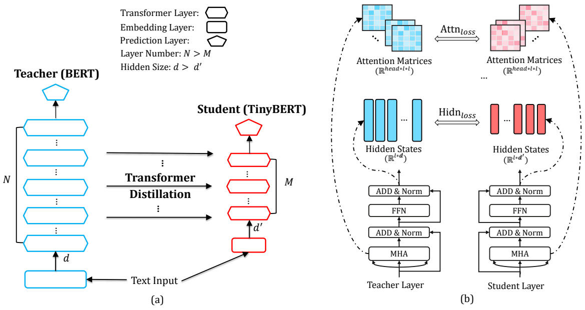

Figure 1: An overview of distillation from BERT to TinyBERT, the figure is taken from the original paper (Jiao et al., 2019) : (a) the general idea of Transformer distillation, (b) the details of Transformer-layer distillation

called "teacher" model as targets. This method combines two losses, namely $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{KD}}$. With λ being a hyper-parameter to control the relative influence of the teacher knowledge transfer.

$$\mathcal{L}_{\text{OKD}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{KD}}. \tag{1}$$

L_{KD}, Knowledge Distillation loss component, is a metric of proximity between logits of the teacher and student models (z^T and z^S respectively). In this paper, we use the Cross-Entropy variation as this loss:

$$\mathcal{L}_{\text{KD}} = -\texttt{softmax}(\boldsymbol{z}^T/\,t)\cdot\texttt{log\_softmax}(\boldsymbol{z}^S/\,t), \tag{2}$$

where t is the softmax temperature applied at training time. At the student model inference its softmax temperature is set to 1. L_{CE} is a classic Cross-Entropy loss for label prediction. The following models are modifying this process each in its own way.

3.2 TinyBERT

The TinyBERT (Jiao et al., 2019) model has the same general architecture as BERT (Devlin et al., 2018), but has fewer layers and smaller hidden and feed-forward sizes. We experiment with the smallest 4-layer model. The number of attention heads on each Transformer (Vaswani et al., 2017) layer is the same as in BERT (12 heads).

The TinyBERT distillation process involves several loss functions. Assume that the student model has $M + 2$ layers, with 0 and $M + 1$ being the indices of the Embedding layer and Prediction Layer respectively. The Transformer (Vaswani et al., 2017) layers are numbered from 1 to M. Every Transformer layer of the student model receives knowledge from the teacher network. Illustrations to this process are presented in Fig. 1. The mapping $g(k)$ between the teacher and student layers is established by a uniform function, in our case the k-th layer of the student model learns from $g(k) = 3k$-th layer of the teacher network (BERT$_{\text{BASE}}$). The objective is defined as MSE between the Attention score matrices plus MSE between the outputs of the Transformer layer (after its FFN part). In order for the dimensions of the student and the teacher hidden states to match, a learnable linear transformation is applied to the student states. The resulting loss function looks as follows:

$$\begin{aligned}\mathcal{L}_{\text{Transformer}}(k) &= \texttt{MSE}(\boldsymbol{H}_k^S \boldsymbol{W}_H \boldsymbol{H}_{g(k)}^T) + \\ &+ \frac{1}{h}\sum_{i=1}^{h} \texttt{MSE}(\boldsymbol{A}_{k,i}^S, \boldsymbol{A}_{g(k),i}^T),\end{aligned} \tag{3}$$

where $\boldsymbol{H}_k$ is the output of the k-th Transformer Layer, $\boldsymbol{W}_H$ is the linear transformation matrix, h is the number of attention heads, and $\boldsymbol{A}_{k,i}$ is the Attention matrix of layer k and head i. Whether the states are associated with the student or the teacher is indicated by the upper indices S and T, respectively. Similarly, MSE is used to distill the embeddings $\boldsymbol{E}^{\{S,T\}}$:

$$\mathcal{L}_{\text{emb}} = \texttt{MSE}(\boldsymbol{E}^S \boldsymbol{W}_e, \boldsymbol{E}^T). \tag{4}$$

For the prediction layer, the Knowledge Distillation loss (2) described above with temperature $t = 1$ is used to adjust the weights of the student.

The TinyBERT distillation process includes two stages. During the first stage called General Distillation (GD), the large unlabeled corpus (English Wikipedia) is used and the general linguistic knowledge contained in model weights is transferred from teacher to student (the prediction layer is untapped). The following objective is minimized during the distillation process:

$$\mathcal{L}_{\text{tiny}} = \sum_{k=0}^{M} \mathcal{L}_{\text{layer}}(S_k, T_{g(k)}). \tag{5}$$

For each layer the loss function is defined by

$$\mathcal{L}_{\text{layer}}(k) = \begin{cases} \mathcal{L}_{\text{emb}}, & k = 0, \\ \mathcal{L}_{\text{Transformer}}(k), & 0 < k \leq M. \end{cases} \tag{6}$$

The second stage is called Task-Specific Distillation (TD) and aims to transfer the task-specific knowledge. It is in fact split into two phases. First, Intermediate Layers Distillation (ILD) is performed with the same loss $\mathcal{L}_{\text{tiny}}$ (5) as in the General Distillation. Then, Prediction Layer Distillation (PLD) is performed with $\mathcal{L}_{KD}$.

The authors apply an augmentation procedure to extend the task-specific training dataset. For every example in the training dataset, N new examples are generated by replacing random words in a sentence with candidates provided by BERT (Devlin et al., 2018) as a language model or by nearest neighbors search in GloVe (Pennington et al., 2014) embedding space. The number N is called the augmentation factor. A detailed description of the algorithm can be found in the original paper (Jiao et al., 2019).

3.3 DistilBERT

The DistilBERT (Sanh et al., 2019) model also shares the general architecture with BERT, slightly modifying it by removing the token-type embeddings and the pooling layer. Following the original work, we use 6-layer DistilBERT. The student layers are initialized directly from the teacher network ($\text{BERT}_{\text{BASE}}$) weights using the uniform strategy for layer mapping.

In the original work, DistilBERT only obtains knowledge from the teacher BERT through General Distillation, although the authors mention experiments with Task-Specific Distillation on SQuAD dataset (Rajpurkar et al., 2016). For General Distillation, a concatenation of English Wikipedia and Toronto Book Corpus (Zhu et al., 2015) is used as training data. The student model uses both supervised training loss (Masked Language Modeling loss, $\mathcal{L}_{\text{MLM}}$) and Knowledge Distillation loss, with logits for the teacher and the student being obtained on Masked Language Modeling (Devlin et al., 2018) task. In addition, the cosine embedding loss $\mathcal{L}_{\text{cos}}$ is used, where the cosine distance is calculated between the outputs of the FFN on the last Transformer layers of the teacher and student. Thus, the general DistilBERT model is trained using the following loss:

$$\mathcal{L}_{\text{distil}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{cos}}. \tag{7}$$

In the original work, the model is then directly fine-tuned on downstream tasks without the help of a teacher network. In the present work, we experiment with the Task-Specific Distillation applied to DistilBERT. We adopt the two-stage procedure similar to TinyBERT. After obtaining the general model, we perform distillation on task-specific datasets. We experimented with different loss function combinations and found out that the best task-specific performance is achieved when three loss functions are used:

$$\mathcal{L}_{\text{distil}}^{\text{TS}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{cos}}, \tag{8}$$

where $\mathcal{L}_{\text{CE}}$ is the standard Cross-Entropy loss (calculated with ground truth data labels) and $\mathcal{L}_{\text{cos}}$ is the same cosine embedding loss as used during the General Distillation. Unlike TinyBERT, we do not split the Task-Specific Distillation stage into two phases (since there is no need to transfer knowledge between deep layers of the networks) and do not use data augmentation.

3.4 ResNet

We also tested our method in application to a computer vision task. We use the classic ResNet model described in (He et al., 2016). The key idea behind the ResNet architecture is a residual block which consists of convolutions layers with skip connections, that helps to reduce the gradient vanishing problem which makes possible to build a deeper neural network.

The key difference between knowledge distillation in NLP (both TinyBERT and DistilBERT variations) and CV is the stage of distillation. A CV distillation does not contain the General Distillation phase, hence teacher and student models are not pre-trained on a general task. In our experiments teacher and student models were ResNet variants, namely ResNet-110 and ResNet-20 respectively.

3.5 Multiple Teachers (Our Method)

A possible way to provide a student model with more knowledge is to make use of multiple teacher models. This can be achieved by combining predictions, outputs, or hidden states of several models. In the present work, we focus on averaging the logits of all teachers before the final softmax layer. That means that multiple teachers are used exclusively during Prediction Layer Distillation to TinyBERT and during Task-Specific Distillation to DistilBERT. All other stages are conducted with one (*primary*) teacher. We also use outputs from only one *primary* teacher model for $\mathcal{L}_{\text{cos}}$ loss function during Task-Specific Distillation to DistilBERT to ensure more fair comparison with TinyBERT. Thus, the use of k teachers $\{S_1, \ldots, S_k\}$ is introduced by slightly changing the L_{KD} formula (2):

$$\mathcal{L}^k_{\mathrm{KD}} = -\mathtt{softmax}(\boldsymbol{z}^T/\,t)\times \\ \times\mathtt{log\_softmax}(\sum_{i=1}^{k}\boldsymbol{z}^{S_i}/\,(k\cdot t)). \tag{9}$$

In this paper we obtain different teacher networks for each downstream task simply by fine-tuning BERT$_{\mathrm{BASE}}$ with different random seeds. We leave the study of other ways of selecting teacher networks to combine as future work.

As for ResNet distillation, since there are no other phases, except the phase of target task distillation, we use multiple teacher distillation technique on it. We again simply fine-tune ResNet-110 with different random seeds to build a set of teachers.

4 Datasets

For evaluation we use a subset of tasks from the GLUE benchmark (Wang et al., 2019) for NLP models. We chose the CoLA task, since the performance drop is the biggest on this dataset for both considered models. The MRPC and SST tasks were chosen in addition to CoLA task due to the average performance drop. Also, the size of MRPC is comparable to CoLA, while SST-2 is a much bigger corpus. Another important feature is that MRPC and SST-2 corpora have test labels publicly available, while the CoLA dataset has not, so below all the results are provided on the development set from this dataset. A brief description of the datasets is provided in this section. We summarize information about the datasets in Table 1. The original results of considered NLP models are presented in Table 2. There are additional datasets in GLUE benchmark, namely: MNLI-m, MNLI-mm, QQP, QNLI, RTE, and STS-B. We provide results on these datasets for reference.

For a computer vision model we chose the classic CIFAR-10 dataset. For the chosen implementation the results, on this dataset are 93.68% and 91.73% for teacher and student respectively[1].

MRPC

The Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) contains sentence pairs in the online news domain. The task is to classify whether the sentences in the pair are semantically equivalent (i.e. have the same meaning). We use classification accuracy as the evaluation metric. The test labels are publicly available for this dataset.

Table 1: Dataset statistics. In “Samples” column we provide train/validation/test split size. “#Tokens“ column contains an average number of words and punctuation marks in a dataset sample. For MRPC we provide the summed number of tokens for a pair of sentences as a sample.

Corpus	Samples	#Tokens	Metric
MRPC	3668 / 408 / 1725	43.9 / 44.0 / 43.5	Accuracy
CoLA	8551 / 1043 / 1063	8.9 / 9.3 / 9.1	MCC
SST-2	67349 / 872 / 1821	9.4 / 19.5 / 19.2	Accuracy
CIFAR-10	50000 / - / 10000	N/A	Accuracy

CoLA

The Corpus of Linguistic Acceptability (Warstadt et al., 2019) consists of sentences from linguistic literature. Each example is annotated with a binary label of whether it is a grammatically acceptable English sentence. We evaluate the Matthews correlation coefficient (Matthews, 1975) on the development set only, due to the test labels are not publicly available.

SST-2

The Stanford Sentiment Treebank (Socher et al., 2013) contains sentences from the movie reviews. We evaluate the classification accuracy on binary sentiment annotation (*positive/negative*), which can be obtained from publicly available fine-grained five-way sentiment labels for both development and test sets.

CIFAR-10

CIFAR-10 dataset was presented in (Krizhevsky et al., 2009). It consists of the 50000 training images and 10000 test images scraped from the Internet. They are labeled with 10 categories for classification: *plane, car, cat, dog, bird, deer, frog, horse, ship, truck*. In this paper, we evaluate the classification accuracy metric in this task. The test labels are publicly available for this dataset.

5 Experiments

In this section we describe the experimental setup and then provide the results and their analysis. All the experiments were performed on a single GeForce RTX 2080 Ti.

[1] These results are better than reported in the original paper (He et al., 2016) due to mistakes in the original implementation.

Table 2: Results are evaluated on the test set of GLUE official benchmark datasets. All models are learned in a single-task manner. In the parentheses we provide performance drop (or gain with '-') in comparison to the teacher model marked with *.

Dataset / **Model**	MNLI-m	MNLI-mm	QQP	**SST-2**	QNLI	**MRPC**	RTE	**CoLA**	STS-B
$\mathrm{BERT}_{\mathrm{BASE}}$*	83.9	83.4	71.1	93.4	90.9	87.5	67.0	52.8	85.2
$\mathrm{BERT}_{\mathrm{SMALL}}$	75.4	74.9	66.5	87.6	84.8	83.2	62.6	19.5	77.1
DistilBERT	78.9 (5.0)	78.0 (5.4)	68.5 (2.6)	91.4 (2.0)	85.2 (5.7)	82.4 (5.3)	54.1 (12.9)	32.8 (20.0)	76.1 (9.1)
TinyBERT	82.5 (1.4)	81.8 (1.6)	71.3 (-0.2)	92.6 (0.8)	87.7 (3.2)	86.4 (1.1)	62.9 (4.1)	43.3 (9.5)	79.9 (5.3)

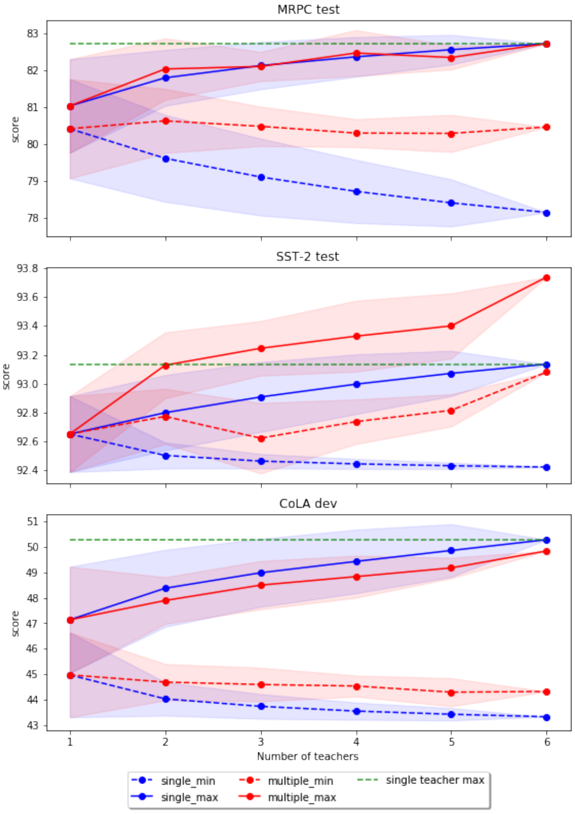

Figure 2: Results for TinyBERT. Shaded: ± one standard deviation.

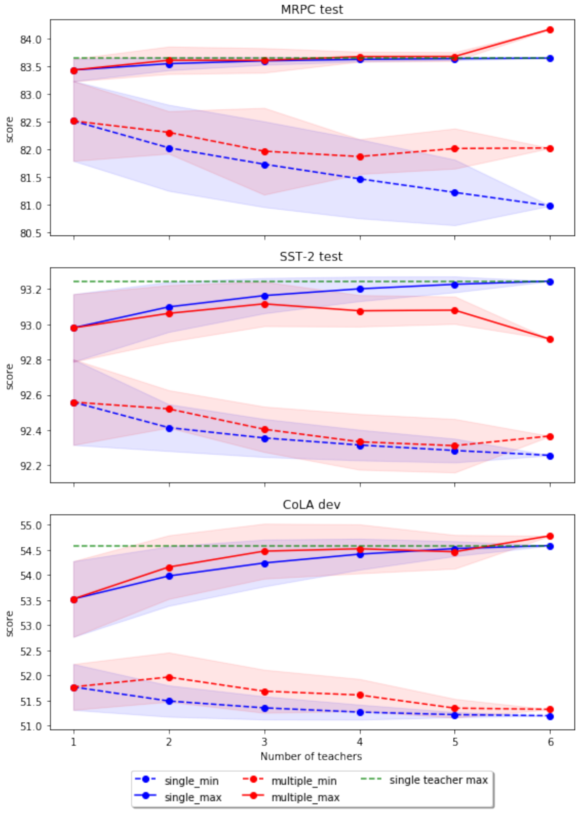

Figure 3: Results for DistilBERT. Shaded: ± one standard deviation.

5.1 Distillation Setup

For each dataset we fine-tuned 6 teacher models with different random seeds. Each teacher NLP model is initialized with $\mathrm{BERT}_{\mathrm{BASE}}$ uncased version from Huggingface's Transformers open-source library[2] (Wolf et al., 2020). We used 30552 as the vocabulary size. Each teacher CV model is initialized as a ResNet-110 model trained on CIFAR-10.

To study the dependence between the number of teachers and the student model scores, we vary the number of teachers from 1 (single teacher distillation) to 6. For each number k, we perform experiments with every k-combination (unordered) from a 6-teacher set. For instance, for $k = 2$ we have $C_6^2 = 15$ possible combinations. Since distillation procedures for NLP models contain parts where a single teacher is used (ILD for TinyBERT and $\mathcal{L}_{\cos}$ for DistilBERT), we actually conduct k experiments for each k-combination with every teacher from that combination being selected as *primary*.

For both TinyBERT and DistilBERT, we experiment only with Task-Specific Distillation. As initialization, general models published by the authors are used[3]. For ResNet models we use an existing

[2] We used Transformers version 2.9.0

[3] General 4layer-312dim *TinyBERT* from https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/

implementation[4] to train the models on the CIFAR-10 dataset, since there is no pre-training stage in this model distillation process.

For TinyBERT, we perform Task-Specific Distillation from a single teacher using the original pipeline presented in (Jiao et al., 2019). The only difference is that for the MRPC dataset where a pair of sentences is passed as model input we apply augmentation procedure to both input sentences simultaneously, while (Jiao et al., 2019) leave one of the sentences unchanged. For distillation from multiple teachers, the Intermediate Layers Distillation part remains the same and Prediction Layer Distillation is modified as described above. For each teacher combination, we perform distillation 3 times on MRPC and CoLA with training data files generated by different runs of the augmentation procedure. Since SST-2 has significantly more training data, the results are less dependent on randomness in the augmentation procedure, so we use only one generated file for it.

For DistilBERT, we use our Task-Specific Distillation procedure described above. As we already mentioned, we do not use data augmentation, so we perform distillation 3 times with different random seeds on all datasets to reduce the impact of randomness on the results of our experiments and to have the same number of experiments as with TinyBERT.

5.2 Results

We conducted a series of experiments in order to prove a hypothesis that a distilled model learned from multiple teachers is more robust to a seed choice. We call a model more robust if it has higher worst possible scores, while keeping the best possible scores about the same level.

At first we would like to compare single-teacher models with multiple teacher ones. To do that, we calculate the minimum and the maximum score achieved with each teacher k-combination. In single teacher mode we simply reuse scores obtained with each teacher included in the combination, while in multiple teacher mode we use all teachers in the combination for the $\mathcal{L}_{\text{KD}}^{k}$. Then we average these minimum and maximum scores over all

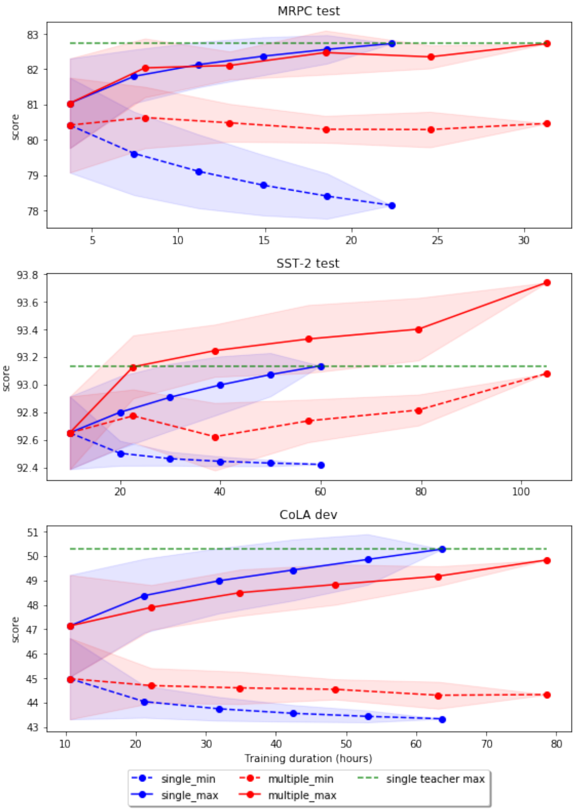

Figure 4: Results for TinyBERT considering time spent on distillation. Shaded: ± one standard deviation.

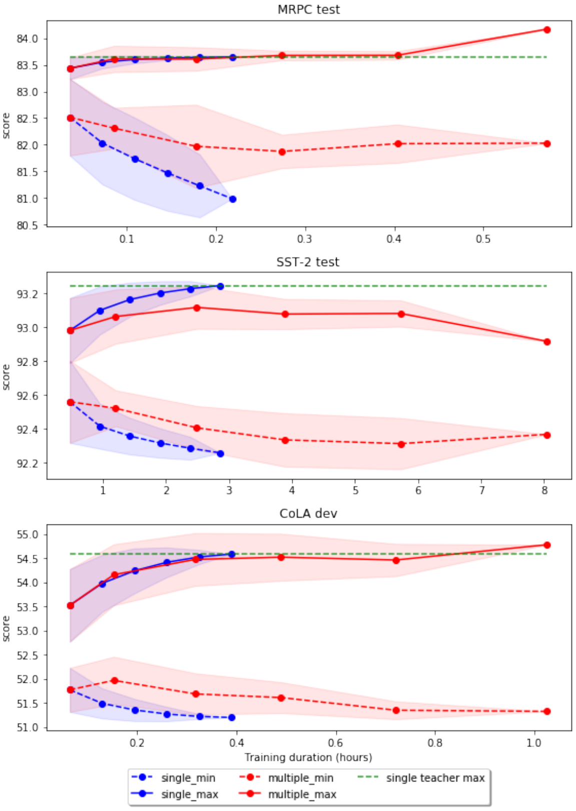

Figure 5: Results for DistilBERT considering time spent on distillation. Shaded: ± one standard deviation.

TinyBERT
distilbert-base-uncased model from Transformers library (https://github.com/huggingface/transformers/tree/master/examples/distillation)

[4] https://github.com/akamaster/pytorch_resnet_cifar10

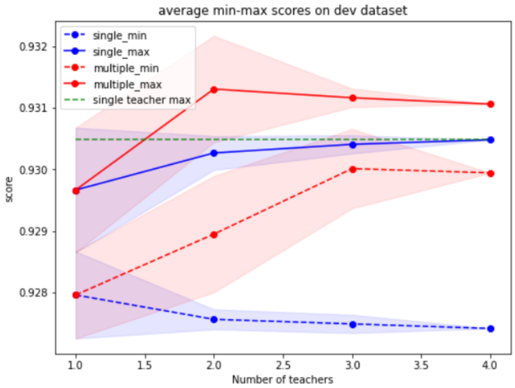

Figure 6: Results for ResNet on CIFAR-10 dataset. Shaded: ± one standard deviation.

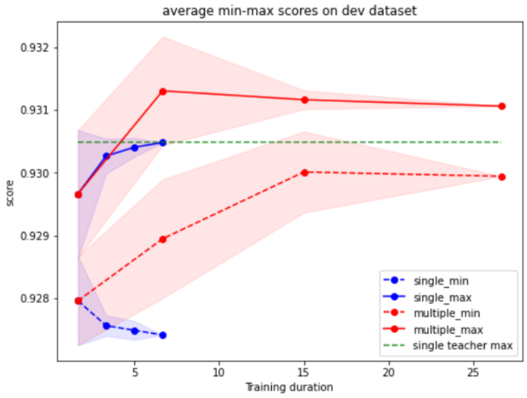

Figure 7: Training duration for ResNet on CIFAR-10 dataset. Shaded: ± one standard deviation.

combinations for each k, obtaining the aggregated measure of models performance.

The scoring results for TinyBERT are provided at Fig. 2, the scoring results for DistilBERT are provided at Fig. 3, while ResNet results are presented at Fig. 6. As one could see the initial hypothesis could be considered true for all the datasets and more than that, the best achievable results are more probable with multiple teacher models for the most models and datasets, with exception of SST-2 for DistilBERT and CoLA for TinyBERT.

One could point out concern regarding the multiple teacher models: these models require significantly larger computational resources to be trained. In order to reduce the potential carbon footprint, we collected additional data for training duration. Since all the training procedures were performed on the same hardware, these measurements could be used for the computational budget comparison. The metrics for TinyBERT are presented at Fig. 4, the metrics for DistilBERT are presented at Fig. 5, while the time consumption for ResNet is presented at Fig. 7. It is readable from the figures that with additional restriction on comparable computational time the hypothesis keeps true, the distilled models are better in the worst case and keep about the same results in the best case, which allows us to call them more robust than the single ones.

6 Conclusion

We showed that the existing distillation process could be improved with the usage of multiple teachers which differ only with random seed initialization. For the NLP models we applied our method to the task-specific distillation, thus improving TinyBERT results. For DistilBERT we modified the original procedure, which led to the improvement in most cases. We also applied the proposed method to ResNet model distillation on the CIFAR-10 task, which led to the quality improvement in all evaluated cases. More than that, the models with roughly the same time consumed by the learning (and distillation) process are better in the worst case, keeping the best results about same level. This keeps true for all the evaluated models and datasets.

As future work, we see an application of the developed technique to the wider variety of models, including the computer vision ones. We hope that our approach can improve the robustness of other modern distillation methods. The additional experiments could be done with more specific tasks, like dialog generation and information retrieval. We hope that our work will foster the research on this topic in the future.

References

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

J. H. Cho and B. Hariharan. 2019. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4793–4801.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

T. Fukuda, Masayuki Suzuki, Gakuto Kurata, S. Thomas, Jia Cui, and B. Ramabhadran. 2017. Efficient knowledge distillation from an ensemble of teachers. In *INTERSPEECH*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. *Proceedings of NeurIPS 2014 Deep Learning Workshop*.

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.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

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

Yuang Liu, Wei Zhang, and Jun Wang. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106 – 113.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pages 2383–2392.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2017. Knowledge adaptation: Teaching to adapt.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

Bharat Bhusan Sau and V. Balasubramanian. 2016. Deep model compression: Distilling knowledge from noisy teachers. *ArXiv*, abs/1610.09650.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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(EMNLP)*.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems(NeurIPS)*, pages 5998–6008.

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 the Proceedings of ICLR.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

M. Wu, C. Chiu, and K. Wu. 2019. Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2202–2206.

Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 690–698.

Shan You, C. Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.

Yang Ze, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. pages 690–698.

Chenrui Zhang and Yuxin Peng. 2018. Better and faster: Knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA. IEEE Computer Society.