2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Improved Knowledge Distillation via Teacher Assistants for Sentiment Analysis

Ximing Dong
Department of Computer Science
University of Manitoba
Winnipeg, Canada
dongxx1104@gmail.com

Olive Huang
Department of Computer Science
University of Auckland
Auckland, New Zealand
ohua281@aucklanduni.ac.nz

Parimala Thulasiraman
Department of Computer Science
University of Manitoba
Winnipeg, Canada
Parimala.Thulasiraman@umanitoba.ca

Aniket Mahanti
Department of Computer Science
University of Auckland
Auckland, New Zealand
a.mahanti@auckland.ac.nz

***Abstract*—Bidirectional Encoder Representations from Transformers (BERT) has achieved state-of-the-art results on various NLP tasks. However, the size of BERT makes application in time-sensitive scenarios challenging. There are lines of research compressing BERT through different techniques and Knowledge Distillation (KD) is the most popular. Nevertheless, more recent studies challenge the effectiveness of KD from an arbitrarily large teacher model. So far, research on the negative impact of the teacher-student gap on the effectiveness of knowledge transfer has been confined mainly to computer vision. Additionally, those researches were limited to distillations between teachers and students with similar model architectures. To fill the gap in the literature, we implemented a teacher assistant (TA) model lying between a fine-tuned BERT model and non-transformer-based machine learning models, including CNN and Bi-LSTM, for sentiment analysis. We have shown that teaching-assistant-facilitated KD outperformed traditional KD while maintaining a competitive inference efficiency. In particular, a well-designed CNN model could retain 97% of BERT's performance while being 1410x smaller for sentiment analysis. We have also found that BERT is not necessarily a better teacher model than non-transformer-based neural networks.**

***Index Terms*—sentiment analysis, BERT, knowledge distillation, teacher assistant, teacher-student network**

I. Introduction

Pre-trained language modelling has achieved state-of-the-art results in various downstream Natural Language Processing (NLP) tasks. As proposed by Devlin et al [1] in 2018, the Bidirectional Encoders Representations from Transformers (BERT) model was pre-trained on Masked Language Modelling and Next Sentence Prediction. The pre-trained BERT can then be fine-tuned for downstream NLP tasks. While BERT based models have been shown to outperform traditional machine learning models, their training is computationally expensive. Moreover, the prolonged inference time makes its application difficult in systems requiring real-time responses, such as web search engines [2]. The size of the model drives research on its compression.

Model compression techniques preceded the emergence of large language models. Recent studies on compressing deep learning models for text were outlined by [3]. In that survey, six types of model compression techniques were discussed, one of which is knowledge distillation (KD) [4], [5]. KD (aka student-teacher network) methods are the most popular model compression technique for Transformer based models. The idea of distilling from Transformer based models is that the teacher model will guide the training of a lightweight student model. In that way, the student model can achieve performance on par with the teacher. The next part of this paper gives a more detailed account of knowledge distillation for NLP.

Despite pivotal research showing significant improvement in accuracy versus model size trade-off through KD, more recent studies challenge the effectiveness of KD from an arbitrarily large teacher [6]–[8]. Also, Mirzadeh et al. [9], via an experiment with CNNs of different sizes for image classification, demonstrated that the student performance degraded as the teacher-student gap increased. To solve this problem, Mirzadeh et al. deployed an intermediate model as a teaching assistant and improved the accuracy of the student network. Whereas, [6] and [7] approached the issue differently by focusing on the training of teacher models. Both approaches are intuitive and reflective of real-life classroom learning scenarios.

Thus far, research on minimising knowledge loss caused by the gap between teacher and student models has been limited to computer vision. Additionally, it has been limited to distillation between teachers and students with similar model architectures. To fill the gap in the literature, we proposed deploying a teacher assistant model, which is an intermediate model that lies between a BERT model and non-transformer-based machine learning models, including CNN and Bi-LSTM, for sentiment analysis. The teacher assistant will be trained with knowledge distilled from BERT first. The trained teacher assistant will then guide the student models, smoothing the knowledge transfer process. The contribution of the paper is pointed out below. We have shown

- that teaching-assistant-facilitated KD significantly outperformed vanilla KD on sentiment classification when a student model is a lot weaker than a teacher model.
- that the effect of KD on student accuracy is minimal or even negative when the performance gap between the teacher model and student model is considerable (in our

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

case, the gap is 8%).
- that, compared to non-transformer-based neural networks such as BiLSTM and CNN, BERT is not necessarily a better teacher for KD.
- that, via our approach, a well-designed CNN retained 97% of BERT's performance while being 1410x smaller.
- that when the student model per se is competent enough for a task, distillation will only provide surplus knowledge or even noise, which would not enhance student performance.

The rest of this paper is organised as follows. Section II discusses the literature on BERT distillation and its drawbacks. Our proposed distillation scheme and model architectures are discussed in Section III. The experimental setup is outlined in Section IV and Section V discusses the results. Section VI concludes this paper with future work.

II. Related Works

This part of the paper analyses in greater detail the literature on deep learning neural networks for sentiment analysis, knowledge distillation from BERT and its drawbacks.

Convolutional Neural Network (CNN), best known for its application in image classification, was shown to have the capability to perform NLP tasks by Kim in [10]. Inspired by Kim's innovative multi-kernel design, Yenter and Verma [11] feeds the output of kernels combined through multiple branches of CNN to an LSTM layer to perform sentiment analysis on IMDB Movie Review Dataset [12]. In our experiment, we implemented only the CNN component, with one convolutional layer and three kernels of different sizes.

Bi-directional Long Short-Term Memory (Bi-LSTM) is a variant of Recurrent Neural Network (RNN). There are three gates in an LSTM cell to remove old information, store new information and store the output of the cell. Lu and Shi in [13] stacked two attention-based Bi-LSTM blocks for document-level semantic classification. In our experiment, we used a simple Bi-LSTM model with one vanilla BiLSTM block to avoid unnecessary complexity in the student model.

Knowledge distillation (KD) is a useful technique for compressing BERT. The seminal study of Hinton et al. [5] proposed what is now considered the traditional KD framework. Hinton et al.'s KD involves a duo objective. The student model will be trained with the guidance of the probability distribution governing all classes (i.e. logits) in a classification task, besides learning from the ground truth labels. There is a trend in recent studies evolving from that fundamental framework and creating more nuanced and complex KD frameworks. For the rest of this section, we will dive into these KD variants.

Patient Knowledge Distillation (PKD): Instead of learning only from the output layer, Sun et al. proposed a Patient Knowledge Distillation (PKD) framework, in which each intermediate layer of the student BERT model learns from some intermediate layers of the teacher BERT model [14]. The results show that student models trained with PKD outperformed vanilla KD. However, a huge performance gap remains between the BERT student model and the BERT teacher model *after* PKD. Moreover, Sun et al. conducted additional experiments to investigate whether having a better teacher will enhance the impact of PKD. For sentiment classification on SST-2 Dataset, the student model guided by the larger teacher model via KD results in slightly worse performance than that guided by the smaller teacher model via KD. Hence, a stronger teacher is not necessarily a better knowledge transferor via vanilla KD. Nevertheless, the settings of Sun et al's experiment do not allow this comparison to be drawn for PKD. Hence, it left unanswered whether PKD could handle distillation from a larger teacher better than KD.

Distil BERT with Augmented Data: [15] argued that text editing (e.g. random insertion and synonym replacement), a technique to augment data, is simple and effective for mitigating the performance degradation. The experiment showed that the student BERT model (with 6 transformers) retained 98.6% of BERT base's performance on SST-2's test set. However, the same student model without text editing has already been shown to retain 98.4% of BERT's performance on the same dataset in [14]. It is doubtful whether this slight additional improvement (by 0.2%) in retention rate through text editing would survive when a much smaller model is used.

KD with Mutual Learning: Some researchers contend that mutual learning of a cohort of simple student networks outperforms distillation from a powerful, well-defined, but static teacher [16], [17]. Zhang et al. in [16] found that a cohort of students learning both from the teacher network and their peers led to significant improvement in image classification from student models being trained independently with KD from a large teacher model. As a variant, Anil et al [17]. suggested having the students learn independently first before performing mutual learning as in Zhang et al's. These researches challenge the conventional perception that effective KD has to distil from a large and powerful teacher model.

KD with Student-Aware Teacher: Other researchers argue that the teacher model should develop some awareness of its students to be a good teacher [6], [7]. In these papers, the authors critique traditional KD for its sole aim of optimising the teacher model for its own inference performance. Whereas, its ability to transfer knowledge is neglected at the fine-tuning stage. To address this problem, Park et al. in [6] proposed a student-friendly teacher network (SFTN). In SFTN, with the advantage of CNN, the teacher model and the student model are modularised into blocks. SFTN combined student branches with the teacher model and trained both parts. By minimising the differences in the representations between the teacher and the student branches, the teacher develops awareness of the student's performance and adjusts itself accordingly. Motivated by the same idea, Zhou et al. in [7] implemented a mechanism that allows a BERT teacher model to receive feedback from a BERT student model. The feedback is the KD loss when distilling the teacher to the student on a separate reserved data split from the original training set. This feedback is used to calculate the second derivatives and perform gradient descent. This approach adapts the knowledge transfer *process* to the student's capacity whereas Park et al. seeks to align the teacher's knowledge *representations* to that of the students. That said, both strongly focus on the teacher's training and adjusting to close the teacher-student gap.

Teaching Assistant KD: It was suggested that having

an intermediate model as a teaching assistant model would improve students' performance. According to Mirzadeh et al.'s experiment in [9], a student model can receive knowledge effectively from a teacher model up to a certain size. KD is futile for teacher models greater than that size. In their experiment, a plain CNN student with 2 convolutional layers is trained via KD with larger teachers of size, 4, 6, 8 and 10 for image classification. Results show that as teacher size increases, the teacher's performance improves while the student's performance starts to drop when the teacher size goes over a particular point. In light of that, Mirzadeh et al. proposed Teacher Assistant Knowledge Distillation (TAKD). The TA network lies somewhere between the teacher and the student networks in terms of its size and capacity. The TA will first be trained with distillation from the teacher network. The trained TA will then be distilled into a student network. The results show that any size of TA improves student's accuracy but the amount of improvement varies compared to vanilla KD. Ganta et al. in [8] used a weighted ensemble of teacher assistants to improve from the single teacher-assistant model. TAs with 8, 7, 6, 5 and 4 are used. The weights of the TAs are optimised for prediction. The results reported indicate further improvement from vanilla KD.

The ideas of student-aware teachers and teaching assistants address effectively the issue of performance degradation due to large model gaps. Both approaches are intuitive and reflective of real-life classroom learning scenarios. [6] and [7] approached the issue by focusing on the training of teacher models, which usually is computationally expensive. Approaching the issue differently, [9] and [8] deployed an extra model that is smaller than the teacher model, requiring fewer computer resources.

In conclusion, the literature shows a pattern that for KD to be effective, more data, more in-depth knowledge, or a smaller teacher-student gap is required. Given the large gap between a BERT model and shallow CNN and Bi-LSTM models, the architectural differences between the teacher and student models and the limited computation resources available, we implemented the TAKD framework for sentiment analysis. To the best of our knowledge, we are the first who adopt TAKD for distilling BERT into a non-transformer-based neural network for a downstream NLP task.

III. Methodology

Our approach focused on leveraging TAKD to improve the effectiveness of knowledge transfer from a BERT model to shallow neural networks. In our experiment, we implemented two student models, two teaching assistant models (TAs) and one teacher model. Distillations from the teacher model to TAs and from TAs to students followed Hinton et al.'s KD scheme. More information about the model architectures and distillation scheme is set out below.

Student Models: The two student models implemented are CNN and Bi-LSTM (hereafter CNN_S and BiLSTM_S). For both models, the first layer takes in the input sequences which are then passed to a pre-processing layer for tokenisation and generating a sequence of indices. The embedding layer is initialised with pre-trained word embeddings GloVe [18]. The embedding dimension is 100 and the embedding weights for the student models are *frozen* during training.

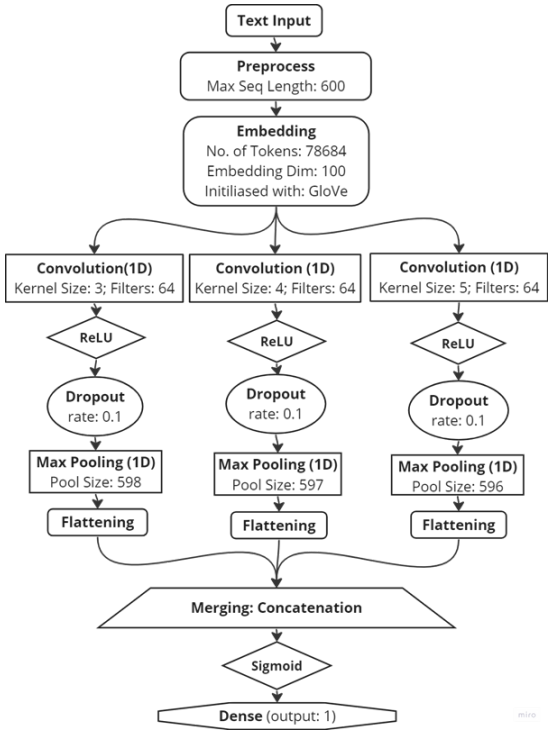

Fig. 1: Diagram of Our Implementation of CNN in [11]. (The diagram is adapted from that in [11])

For CNN (hereafter CNN_S), the embedded sequences are passed into three branches of 1-dimensional convolutional layers with kernel sizes of 3, 4, and 5, respectively (See Fig 1). Using a combination of different kernels simulates the use of a combination of tri-grams, 4-grams and 5-grams, which may better capture the relationships between neighbouring words for sentiment analysis [19]. For each kernel size, there are 64 filters. After applying the ReLU activation on the outputs of CNN layers, a 10% dropout is applied to each branch. Each branch then max-pooled each kernel size of input into a single output. The output of the 1-dimensional layer is then flattened before being concatenated together. The concatenated representation is now passed to the Dense layer for classification into two classes.

Regarding BiLSTM (hereafter, BILSTM_S), the architecture is a standard one. The embedded sequences are passed to two LSTM layers in opposite directions. Each of the LSTM layers has 256 output units. A 10% dropout is applied to the LSTM layers. The outputs of the two LSTM layers are concatenated before they are passed to a Dense layer for classification.

Teaching Assistant Models: The two teaching assistant models are student models with the embedding layer *unfrozen* (hereafter CNN_TA and BiLSTM_TA). Specifically, CNN_S is trained with knowledge distilled from CNN_TA and BiLSTM_S is trained with knowledge distilled from BiLSTM_TA (See Fig 2).

Teacher Model: The teacher model is a BERT base model (hereafter BERT_BASE) with 12 layers of Transformers, a hidden size of 768 and 12 attention heads. We loaded the BERT encoder bert_en_uncased_L-12_H-768_A-12[1] from TensorFlow Hub, as well as the corresponding text preprocessing

[1]https://tfhub.dev/tensorflow/bert_en_uncased_L12_H768_A12/4

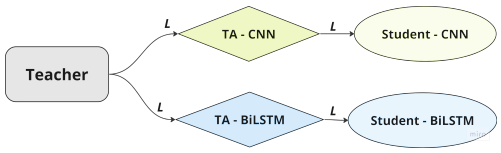

Fig. 2: Diagram Showing the Scheme of TAKD. L denotes the loss function applied to the specific distillation.

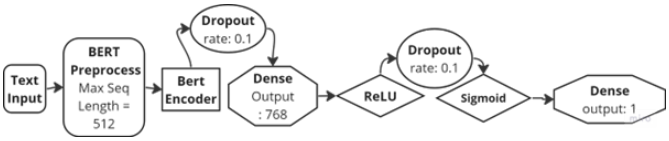

Fig. 3: Diagram of Our Implementation of BERT for Sentiment Classification

model.[2] The default sequence length produced by the said preprocessor is 128. Thus, to extend the sequence length to 512, we subclassed the preprocessor, which allowed us to customise the sequence length. As shown in Fig 3, the processed sequence will then be passed to a BERT encoder. The pooled output of the encoder that represents each input sequence as a whole will be passed to the Dense layer with 768 units. After applying a 10% dropout, the output of the Dense layer will be passed to the final Dense layer for classification. The teacher model is fine-tuned with the cross entropy loss function.

Distillation Objective: We used Hinton et al's distillation framework to distil knowledge from the last layer of $\text{BERT}_{\text{BASE}}$ to the TAs and from the last layer of TAs to their corresponding students. Before distillation, $\text{BERT}_{\text{BASE}}$ is fine-tuned. The distillation objective consists of two parts. The first part penalises the mismatch between a student's output and the ground-truth label (i.e. hard loss). This mismatch is measured by cross-entropy loss in our implementation. The second part encourages the student to learn from a teacher and penalises the mismatch between the student's logits and the teacher's logits (i.e. soft loss). This mismatch is measured by Kullback–Leibler divergence loss in our implementation. The overall distillation objective is a weighted sum of the soft loss ($KL(y_{pred}, y_{teacher})$) and hard loss($CE(y_{pred}, y_{true})$), as represented by (1) below. α controls the weights attached to each loss. Usually, soft loss is attributed more weight. In our experiment, we decided to fix α at 0.1. The same objective function is used for KD from teacher to TAs and from TAs to students (See Fig 2).

$$L = \alpha \cdot CE(y_{pred}, y_{true}) + (1-\alpha) \cdot KL(y_{pred}, y_{teacher}) \quad (1)$$

IV. Experiments

The **experimental environment** is summarised in Table I. The experiment was run on an Ubuntu 20.04.6 LTS 64-bit Operating System with 16 Intel Xeon Processors (Cascade-lake) 2.69 GHz (1 thread per core). A single NVIDIA A100 GPU with 80G of RAM is used.

The **datasets** we used are

- IMDB Movie Review Dataset [12].: The dataset contains 25000 highly polar movie reviews for training and 25,000 reviews for testing. There are two data labels: negative and positive. The dataset was downloaded from the torchnlp.datasets package. 5000 out of the 25000 reviews were reserved for the validation set.
- Stanford Sentiment Treebank 2 (SST-2) [20]: SST-2 is a movie review dataset with two labels indicating positive and negative reviews. The dataset was downloaded from this link.[3] Since the original test set is not labelled, only the train set and the original dev set are used in the experiment. The original train set and the validation set are combined into a single dataset before it is divided into the train, validation and test sets by a ratio of 8:1:1.

Environment Components	Description
Server Configuration	Ubuntu 20.04.6 LTS 64-bit Operating System with 16 Intel Xeon Processors (Cascadelake) 2.69 GHz (1 thread per core)
GPU Configuration	a single NVIDIA A100 GPU with 80G of RAM
Programming Language and Tool	Python 3.8.10 and Jupyter Notebook
Libraries and Frameworks	tensorflow 2.12.0, tensorflow_hub 0.13.0, numpy 1.24.3

TABLE I: Experiment Environment

Datasets	IMDB Movie Review Dataset	SST-2 Dataset
Training Set	20,000	54,576
Validation Set	5,000	6,822
Test Set	20,000	6,823

TABLE II: Summary of Dataset Splits

A summary of the splits of the two datasets is shown in Table II.The reviews in these two datasets are **cleaned** and **preprocessed** before passing them to the model for training and predictions. Particularly, stopwords, punctuation marks, HTML tags, URLs, characters that are not letters or digits and successive whitespaces are removed. The maximum length of sequences differs for both datasets. For the IMDB Movie Review Dataset, we chose 512 to be the maximum sequence length for the teacher model ($\text{BERT}_{\text{BASE}}$ and 600 for the TA and student models. Regarding the SST-2 dataset, the maximum sequence length of the $\text{BERT}_{\text{BASE}}$ model is 512 while it is 28 for the TA and student models. In general, sentences in the SST-2 dataset are a lot shorter than those in the IMDB Movie Review Dataset.

Fine-tuning $\text{BERT}_{\text{BASE}}$ is the very first step of our experiment. As summarised in Table III, $\text{BERT}_{\text{BASE}}$ for the IMDB Dataset is trained for 10 epochs with no early stopping applied. It was found through empirical trials that a batch size of 16 led to the best accuracy on the validation set. The optimiser used is an Adam optimiser with a self-defined warm-up schedule that applies to the learning rate, which is set to be 5e-6 initially. The warm-up schedule has a warm-up learning rate of 0 and 1,250 warm-up steps. The post-warm-up learning rate schedule is a linear decay from the initial learning rate to zero over 12,500 training steps. For all the models, the model that gives the best accuracy on the validation set is used for prediction on the test set. The hyperparameter setup for of the $\text{BERT}_{\text{BASE}}$ model for the SST-2 dataset is similar despite the difference in initial

[2]https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3

[3]https://dl.fbaipublicfiles.com/glue/data/SST2.zip'

#	Models	Batch Size	Initial Lr	Optimiser	Epochs	EarlyStopping
1	$BERT_{BASE}$	16	5e-6 / 2e-5	Customised Adam	10	Not Applied
2	CNN_{TA}	64	0.001	Adam	30	Applied
3	CNN_{S}	64	0.001	Adam	30	Applied
4	CNN_{FT}	64	0.001	Adam	30	Applied
5	$BiLSTM_{TA}$	64	0.001	Adam	40	Applied
6	$BiLSTM_{S}$	64	0.001	Adam	40	Applied
7	$BiLSTM_{FT}$	64	0.001	Adam	40	Applied

TABLE III: Summary of Hyperparameter Setup for IMDB Movie Review and SST-2

learning rate and the batch size. The batch size for the SST-2 dataset is 64 and the initial learning rate is 2e-5.

All the other models (CNN_{TA}, CNN_{S}, CNN_{FT}, $BiLSTM_{TA}$, $BiLSTM_{S}$, $BiLSTM_{FT}$) have very similar setup (See Table III). CNN_{FT} and $BiLSTM_{FT}$ are CNN and BiLSTM baselines fine-tuned from scratch without KD. All of these models are trained with an initial learning rate of 0.001, which is the default initial learning rate of the Adam optimiser. It was decided that BiLSTM should be trained longer because it was observed that they tended to take longer to converge. Therefore, BiLSTM models are trained for 40 epochs and CNN models for 30 epochs. Early stopping was applied to these six models. However, for the IMDB Movie Review, early stopping only starts to operate from the 5th epoch onwards and allows no improvement in the validation accuracy for 6 epochs before the training is terminated. As for the SST-2 dataset, early stopping starts to apply from the 3rd epoch and allows no improvement in validation accuracy for only 4 epochs before the training is terminated. It was found empirically that using an early stopping scheme as generous as that for the IMDB Dataset resulted in sub-optimal model performance on the SST-2 Dataset. Hence, more stringent early stopping rules are applied to encourage the training to stop earlier.

V. Results and Discussion

This section discusses the test accuracy, inference time, model size and the model's ability to retain BERT's performance. Our implementation of BERT has a test accuracy of 90.98%, as shown in Table IV. Although this is lower than expected, we are not concerned with the teacher converging sub-optimally for KD because a perfectly fine-tuned BERT teacher is not necessarily a good teacher for KD, as explained earlier. Both baseline models CNN_{FT} and $BiLSTM_{FT}$ have similar test accuracy, around 82%, even though CNN_{FT} is almost 10 times smaller than $BiLSTM_{FT}$.

Results for the IMDB Movie Review Dataset (Table IV) show that TAKD leads to higher accuracy than vanilla KD. CNN_{S}_TAKD improves from CNN_{S}_KD by around 5.5% whereas CNN_{S}_KD's improvement from the baseline CNN_{FT} is nearly negligible (roughly 0.2%). Similarly, $BiLSTM_{S}$_TAKD improves from $BiLSTM_{S}$_KD by 3.4% whereas $BiLSTM_{S}$_KD performs slightly worse than the baseline $BiLSTM_{FT}$ by roughly 0.3%. Moreover, CNN_{S} is a particularly smart student for TAKD. Through TAKD, it retains around 97% of the teacher's performance (90.98%) while being 1410 times smaller. Also, its inference time is one of the shortest.

However, when the student models are able to perform nearly as well as the teacher, neither KD nor TAKD improves

#	Models	Accuracy (%)		Inf Time (s)		No. Params	Acc Retained (%)	
		IMDB	SST-2	IMDB	SST-2	IMDB | SST-2	IMDB	SST-2
0	$BERT_{BASE}$ (Devlin et al. [1])	-	93.5	-	-	-	-	100
1	$BERT_{BASE}$	90.98	93.48	152	37	108,901,634 (1x)	100	100
2	CNN_{FT}	82.22	91.07	4	1	77,185 (1410x)	90.4	97.4
3	CNN_{S}_KD	82.39	91.13	5	1	77,185 (1410x)	90.6	97.5
4	CNN_{S}_TAKD	87.86	91.25	4	1	77,185 (1410x)	96.6	97.6
5	$BiLSTM_{FT}$	82.31	91.26	25	2	731,649 (149x)	90.5	97.6
6	$BiLSTM_{S}$_KD	82.03	90.94	24	2	731,649 (149x)	90.2	97.2
7	$BiLSTM_{S}$_TAKD	85.45	91.09	25	2	731,649 (149x)	93.9	97.4

TABLE IV: Summary of Main Results for the IMDB Movie Review Dataset and SST-2 Dataset - CNN_{S}_KD and $BiSLTM_{S}$_KD are student models trained with vanilla KD. CNN_{S}_TAKD and $BiLSTM_{S}$_TAKD are student models trained with TAKD. Acc Retained is short for the accuracy retained

students' performance (See Table IV). All the sentences in SST-2 consist of less than 28 tokens after preprocessing and our chosen maximum sequence length is 28. That means the entirety of the sequence was passed to both the student and teacher models for training and prediction. In that respect, compared with the IMDB Dataset, SST-2 is a more manageable dataset to learn from. The student models themselves already handle easy-to-learn knowledge very well. This is shown by the accuracies of CNN_{FT} and $BiLSTM_{FT}$, which are 91.07% and 91.26%, respectively. The accuracies of the student models are only approximately 2% lower than that of the teacher model, which is 93.48%. Despite that, KD from the teacher model to the student models did not improve students' performance, as shown by the accuracies of CNN_{S}_KD and $BiLSTM_{S}$_KD (91.13% and 90.94%, respectively.) Nor did TAKD make a real difference because the accuracies of CNN_{S}_TAKD and $BiLSTM_{S}$_TAKD are 91.25% and 91.09%. What it illustrates is that when the knowledge is relatively easy, neither TAKD nor KD provides additional information to the training of a competent student model.

Having discussed the main findings of our experiment, we now move on to investigate further the impact of teacher-student gap on the effectiveness of knowledge transfer. We have conducted additional experiments for this purpose. The additional experiments involve training the TA models CNN_{TA} and $BiLSTM_{TA}$ independently without KD from $BERT_{BASE}$ and distil knowledge from these independently trained TAs to our student models, CNN_{S} and $BiLSTM_{S}$, respectively. This will allow us to have another teacher to compare with $BERT_{BASE}$. The setup of hyperparameters of the newly created models remains the same as that shown in Table III.

The results in Table V show, for the IMDB Dataset, that standalone TAs, CNN_{TA_FT} and $BiLSTM_{TA\_FT}$, outperform student models trained with vanilla KD by more than 4%. Also, Settings 3 and 6 demonstrate that performance loss is minimal (less than 1%) when distilling the standalone TAs into the students. These students significantly outperformed students distilled from BERT in Settings 1 and 4. Thus, compared with BERT, a simpler model that is similar to but slightly stronger than the student is a better teacher for KD.

Furthermore, comparing Settings 1 and 2, when the teacher model is fixed, knowledge is not transferable to the student when the compression ratio is as high as 1410x. However, when the compression ratio is reduced to 14:1, KD is

Setting	Teacher	Student	Accuracy (%)		Inf Time (s)		No. Params		Acc Retained (%)	
			IMDB	SST-2	IMDB	SST-2	IMDB	SST-2	IMDB	SST-2
N/A		$BERT_{BASE}$	90.98	93.48	152	37	108,901,634 (1x)	108,901,634 (1x)	100	100
N/A		CNN_{TA_FT}	86.97	91.38	4	1	7,945,585 (14x)	1,499,585 (73x)	100	100
N/A		$BiLSTM_{TA\_FT}$	86.74	91.66	24	1	8,600,049 (12x)	2,154,049 (51x)	100	100
1	$BERT_{BASE}$	CNN_S	82.39	91.13	5	1	77,185 (1410x)	77,185 (1410x)	90.6	97.5
2	$BERT_{BASE}$	CNN_{TA}	86.97	91.13	4	1	7,945,585 (14x)	1,499,585 (73x)	95.6	97.5
3	CNN_{TA_FT}	CNN_S	86.32	91.13	4	1	77,185 (1410x)	77,185 (1410x)	99.3	99.7
4	$BERT_{BASE}$	$BiLSTM_S$	82.03	90.94	24	2	731,649 (149x)	731,649 (149x)	90.2	97.2
5	$BERT_{BASE}$	$BiLSTM_{TA}$	87.14	90.99	24	2	8,600,049 (12x)	2,154,049 (51x)	95.8	97.3
6	$BiLSTM_{TA\_FT}$	$BiLSTM_S$	85.92	90.97	25	2	731,649 (149x)	731,649 (149x)	99.1	99.2

TABLE V: Different Size Comparison between Teachers and Students on IMDB Movie Review Dataset and SST-2 Dataset - TAs trained with KD as part of TAKD (CNN_{TA_FT}) and without ($BiLSTM_{TA\_FT}$). Models trained with a teacher with vanilla KD. Acc Retained (Accuracy retatined) is calculated with respect to the actual teacher model used.

effective again. Likewise, comparing Settings 4 and 5, for BiLSTM, KD is futile or even results in worse results when the teacher model is 149 times as great as the student model but becomes effective again when the gap is reduced to 12:1. That suggests that a teacher model can only transfer knowledge to student models down to a certain size.

Table V shows interesting findings on the SST-2 dataset. Settings 1 and 2 suggest that a model with dynamic pre-trained word embeddings is *not* a more competent student than that with static pre-trained word embeddings. Comparing Settings 4 and 5 for the BiLSTM models gives the same conclusion. Presumably, this is because GloVe per se provides enough foundational knowledge for the model to learn a simple dataset *fully*. It follows that additional signals from the teacher model become surplus or even noise.

Comparing Settings 1 and 3, we notice that distilling from a more knowledgeable teacher does not necessarily results in better student performance. With the student model being the same in these two settings, KD from $BERT_{BASE}$, a more knowledgeable teacher, does not provide signals more than CNN_{TA_FT} does. The same conclusion could be drawn for BiLSTM by comparing Settings 4 and 6.

In summary, we found that KD and TAKD are the most effective when the dataset is, to a degree, challenging for the student. When the student is fully competent in extracting all the useful information in a dataset, KD or TAKD would only provide surplus or noise. Nevertheless, when the student itself is not competent enough for the task. KD will provide extra knowledge, provided the capacity gap between the teacher and student model is not overly large. Otherwise, TAKD is necessary for knowledge to be effectively transferred from the teacher to student models.

VI. Conclusion

In this paper, we used a teaching assistant model to reduce the performance gap between a large BERT base model and two shallow non-transformer-based neural networks for sentiment analysis. We have shown that TAKD significantly outperformed KD for sentiment analysis when the compression ratio between the teacher model and the student model is considerable.

For future work, we plan to investigate the impact of the size of the teaching assistant models on the effectiveness of knowledge distillation. Also, we are interested in exploring whether having the student learn from a weighted ensemble of the teaching assistant and the teacher will improve the learning process even further.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Z. Yang, L. Shou, M. Gong, W. Lin, and D. Jiang. 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*, page 690–698. Association for Computing Machinery, 2020.

[3] M. Gupta and P. Agrawal. Compression of deep learning models for text: A survey. *ACM Trans. Knowl. Discov. Data*, 16(4), Jan 2022.

[4] C. Bucila, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 535–541. Association for Computing Machinery, 2006.

[5] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[6] D. Y. Park, M.-H. Cha, C. Jeong, D. Kim, and B. Han. Learning student-friendly teacher networks for knowledge distillation. *CoRR*, abs/2102.07650, 2021.

[7] W. Zhou, C. Xu, and J. J. McAuley. Meta learning for knowledge distillation. *CoRR*, abs/2106.04570, 2021.

[8] D. P. Ganta, H. D. Gupta, and V. S. Sheng. Knowledge distillation via weighted ensemble of teaching assistants. In *2021 IEEE International Conference on Big Knowledge*, pages 30–37, 2021.

[9] S. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:5191–5198, 2020.

[10] Y. Kim. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[11] A. Yenter and A. Verma. Deep CNN-LSTM with combined kernels from multiple branches for imdb review sentiment analysis. In *8th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference*, pages 540–546. IEEE, 2017.

[12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics, 2011.

[13] X. Shi and R. Lu. Attention-based bidirectional hierarchical lstm networks for text semantic classification. In *10th International Conference on Information Technology in Medicine and Education (ITME)*, pages 618–622, 2019.

[14] S. Sun, Y. Cheng, Z. Gan, and J. Liu. Patient knowledge distillation for bert model compression. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[15] S. Sun, D. Yu, and C. Lv. Text editing for augmented distilled bert. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence*, pages 437–442, 2020.

[16] Y. Zhang, T. Xiang, T. Hospedales, and H. Lu. Deep mutual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4328. Institute of Electrical and Electronics Engineers (IEEE), 2018.

[17] R. Anil, G. Pereyra, A. Passos, R. Ormándi, G. E. Dahl, and G. E. Hinton. Large scale distributed neural network training through online distillation. In *6th International Conference on Learning Representations*. OpenReview.net, 2018.

[18] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[19] A. Tripathy, A. Agrawal, and S. Rath. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57, 2016.

[20] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics, 2013.