Learning by Teaching, with Application to Text Classification

Fengqi Yang
University of California, San Diego
fey002@ucsd.edu

1 Introduction

1.1 Research Problem and Importance

We are aiming to optimize the text classification problem by developing and advancing the Learning-by-Teaching framework introduced by Professor Pengtao Xie. Unstructured text is everywhere, such as emails, chat conversations, news, and social media but it's hard to extract value from this data unless it's organized in a certain way. Doing so used to be a difficult and expensive process since it required spending time and resources to manually sort the data or creating handcrafted rules that are difficult to maintain. Our goal is to analyze text and then assign a set of pre-defined tags or categories based on its content. Learning-by-Teaching method with machine learning model could be a great alternative to structure textual data in a fast, cost-effective, and scalable way.

The field of text classification is both incredibly useful and wide-reaching, both from a business and academic approach. Several components of text analysis have already become prevalent tools used in much of industry. One such application is sentiment analysis, in which a classifier identifies the overall emotion associated with a line of text. The classification can be adapted to the whole text, or to a granular level, e.g. to a sentence or even just a phrase. Sentiment analysis is commonly utilized in business, providing the foundation for much of customer-related services or market research.

Ultimately, text classification provides a level of automation and simplification that was not previously available. The process of extracting structure out of unstructured text data can help build/standardize a foundation, augment the user experience, and streamline searching. Applying LBT to the text classification problem can lead to increased accuracy, allowing for more dependability on NLP models.

2 Related Works and Solutions

In this section, we will discuss related previous work on this research problem and propose our solutions together with its novelty and significance.

2.1 Related Works

Text Classification has exploded in popularity, notably in the past 5 years. Many papers have been published in attempts to develop state-of-the-art models. One paper by Kowsari, et al. (2019)[1] succinctly explores a couple of the algorithms being used, and discusses the variable methodologies applied in creating the models, as well as their limitations and applications. He compiles a list of previous papers and details them, providing their feature extraction, novelty, architecture, corpus, validation measure, and limitation. Of note is the paper by Z. Yang, et al. (2019)[2] which covers Hierarchical Attention Networks, but only works for document-level data.

In Devlin, et al. (2019)[10], the study introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. BERT is designed to

Preprint. Under review.

pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks.

Sanh et al. [3] proposed a method to pre-train a smaller general purpose language representation model, called DistilBERT, which can then be fine tuned with good performances on a wide range of tasks like its larger counterparts. With self-distillation, the student and teacher models can benefit from each other. Our proposed strategies are orthogonal to the approaches with external data and knowledge. The distillation loss can also be regarded as a regularization to improve the generalization ability of the model.By leveraging knowledge distillation during the pre-training phase, it can reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. The self-ensemble method with parameter averaging can improve BERT without significantly decreasing the training efficiency.

Finally, Xie, et al. (2020)[4] introduces Learning by Teaching (LBT), which aims to emulate the successes of human learning behavior and replicate them for machine learning models. In essence, one model (the "teacher") generates a pseudo-labeled dataset, and teachings another model (the "student"). The teacher re-evaluates based on the student's performance on a validation dataset, and optimizes until the performance is high. Xie propose a multi-level optimization framework to formalize LBT which involves three learning stages: the teacher learns; the teacher teaches the student by pseudo-labeling; the teacher improves its architecture based on the validation results of itself and of the student.

2.2 Proposed Solution, Novelty and Significance

The proposed solution is using learning-by-teaching method built on machine learning model to apply on text classification tasks. Specifically in this project, we will study how BERT models can be combined with Learning by Teaching framework. Two BERT models will be used to represent teacher and student respectively. The general idea is we pre-train a model for teacher and then initializing the student's network weight by transferring the teacher's weights to represent the teaching process. After feeding the training data to the student, it will compare the output labels (both cross entropy validation loss and mean squared loss of teacher's output) to perform backward loss minimization. After the student found a updated network weight, it will provide update to the teacher and teacher can update its weights from the "feedback". In this way, those two models could learn from each other and both model can achieve better performance.

In Xie, et al. (2020)[4], the study showed Learning by Teaching (LBT) framework can be applied to Neural Architect Search on image classification tasks. In our newly proposed solution, we will explore the possibilities to apply LBT framework on text classification tasks through state-of-the-arts approaches BERT. If the solution can achieve state-of-the-art performance on benchmark data set, it will demonstrate this LBT method could be applied to text classification effectively. More importantly, by applying the LBT framework, student and teacher models can benefit from each other. The two part loss minimization process can also be regarded as a regularization to improve the generalization ability of the model.

2.3 Limitations and Challenges

There are some potential limitations and challenges for this project. First of all, the algorithm in theory will improve generalization ability of one model at a comparable accuracy performance. Previous related work Sanh et al. [3] was focus on knowledge distillation on student BERT model which has less number of layers without significantly decreasing the training efficiency. In our study for applying LBT framework to improve teacher's learning outcome specifically, it will be challenge to show text classification accuracy can be improved significantly. Secondly, we expect long training time for running BERT model and more time for fine tuning hyper parameter. Because the network of BERT model is relative deep and it has a huge amount of parameters, the training process for each model will be time consuming. Especially the model usually needs over 50000 steps to achieve a

Table 1: Notations in Learning by Teaching

Notation	Meaning
A	Teacher Model
B	Student Model
T	Network weights of the teacher
S	Network weights of the student
D_t^{tr}	Training data of the teacher
D_t^{val}	Validation data of the teacher
D_s^{tr}	Training data of the student
D_s^{val}	Validation data of the student
D_{tl}^{val}	Teacher output on validation dataset
D_{sl}^{val}	Student output on validation dataset

significant improvement on accuracy. Due to the time limit, we might not able to perform experiments on different benchmark data sets, which will potentially diminish the effectiveness of the final result.

3 Method

In this section, we will discuss the methods for the proposed solution.

3.1 Learning by Teaching

In the Learning by Teaching framework, the main framework is using a teacher model and a student model. Both teacher model and student model will be trained on the same target task. Ultimately, the goal for applying LBT framework is to help the teacher model achieve better performance. As Xie, et al. (2020)[4] already proved that LBT can successfully improve image classification in his study, for this project, the target task is text classification.
In our newly proposed algorithm, we utilized a different approach to perform teaching compared to the study of Xie, et al. (2020)[4]. In Xie, et al. (2020)[4], teaching is performed by pseudo-labeling Hinton, et al. (2015a)[5]: the teacher uses its model to generate a pseudo-labeled dataset; the student is trained on the pseudo-labeled dataset, and incorporate two methodologies. Our approach of performing teaching is inspired by knowledge distillation Hinton, et al. (2015a)[5] and Sanh, et al. (2019)[3]: Taking advantage of the common dimensionality between teacher and student networks, we initialize the student from the teacher by taking one layer out of two. In our study, because the focus is applying Learning by Teaching framework to improve learning outcome instead of knowledge distillation, so we make student networks to have same number of layers as teacher. More specifically, we initialized the teacher model by a pre-trained BERT , and the student model is initialized with the pre-trained weights form the teacher. Thus, teaching will be performed through transferring network weights from teacher model to student model instead of pseudo-labeled dataset.
In our newly proposed algorithm, the learning outcome is improved by making the teacher model as an "feedback receiver" of the student model within the student model's fine-tuning stage. Intuitionally, the teacher learned and transfer the knowledge (network weight) to the student. The student will validate itself and provide "feedback" to the teacher for further improvement. Theoretically, the teacher has a learnable BERT model A and a set of learnable network weights T, The student has a learnable BERT model B has and a set of learnable network weights S. The teacher has a training dataset D_t^{tr} and a validation dataset D_t^{val}. The student has a training dataset D_s^{tr} and a validation dataset D_s^{val}. In our framework, both the teacher and student perform learning, which is organized into three stages. In the first stage, the teacher trains its network weights on its training dataset

$$T^*(A) = \min_T L\left(T, A, D_t^{(\mathrm{tr})}\right) \tag{1}$$

Note that teacher model A didn't fine tuning network weights by performing backward propagation in this first stage. Instead, the teacher model proceed to evaluation stage directly. If A is fine tuned by minimizing this validation loss through backward propagation itself, then this will be a trivial solution.
In the second stage, the teacher transfer the current knowledge $T^*(A)$ to student. The teacher

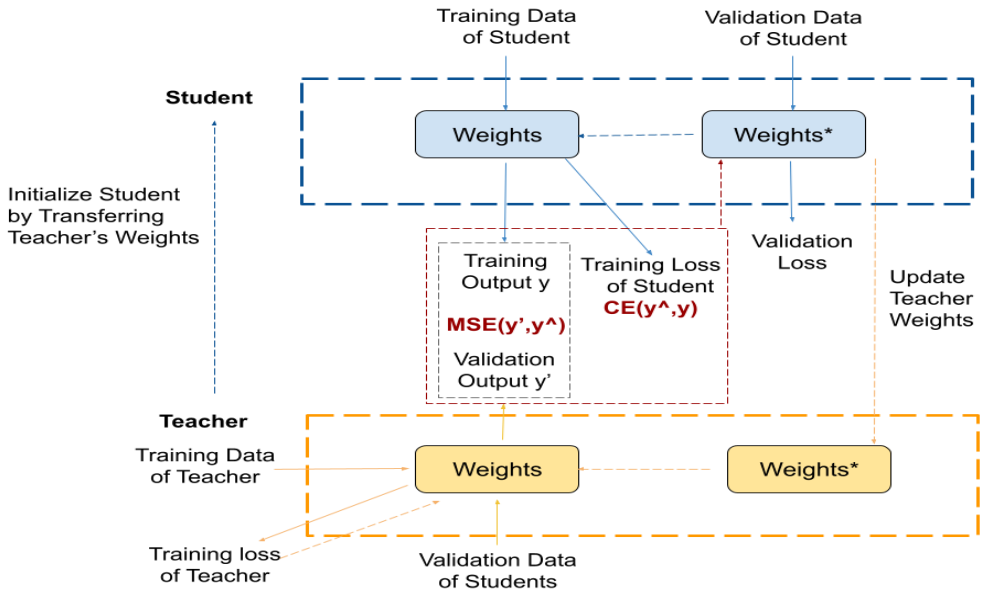

Figure 1: Illustration of learning by teaching on BERT. The solid arrows denote the process of making predictions and calculating losses. The dotted arrows denote the process of updating learnable parameters by minimizing corresponding losses.

validates its network weights $T^*(A)$ on its validation set D_t^{val} and output D_{tl}^{val}. The student validates its network weights $S\left(T^*(A)\right)$ on its validation set D_s^{val} and output D_{sl}^{val}. then the student leverages
1. validation loss together with
2. loss between D_t^{val} and D_s^{val}
to fine tune its network weights S through backward propagation.

$$S^*\left(T^*(A)\right) = \min_S L\left(S, D_s^{val}, T^*(A)\right) + \min \lambda L\left(D_t^{val}, D_s^{val}, T^*(A), S\right) \tag{2}$$

The first part is cross-entropy loss, and second part is mean squared error loss, where λ is a trade off parameter. In the third stage, teacher model update the network weights to T**(A) by subtracting the difference between student's network weights S* and teacher's previous network weights T*(A)

$$T^{**}(A) = T^*(A) - \alpha(T^*(A) - S^*\left(T^*(A)\right)) \tag{3}$$

Similar to Xie, et al. (2020), the three stages are mutually dependent: $T^*(A)$ learned in the first stage is used to define the objective function in the second stage; $T^*(A)$ and $S^*(T^*(A))$ learned in the first two stages are used to define the update function in the third stage; the updated network weights of A in the third stage in turn changes the objective function in the first two stages. Putting these pieces together, we have the following LBT framework, which is a two-level optimization problem plus an update in the final stage:

$$\begin{aligned} T^{**}(A) &= T^*(A) - \alpha(T^*(A) - S^*\left(T^*(A)\right)) \\ s.t. S^*\left(T^*(A)\right) &= \min_S L\left(S, D_s^{val}, T^*(A)\right) + \min \lambda L\left(D_t^{val}, D_s^{val}, T^*(A), S\right) \\ T^*(A) &= \min_T L\left(T, A, D_t^{(\mathrm{tr})}\right) \end{aligned}$$

3.2 Optimization Algorithm

In this section, we will illustrate the algorithm derived to solve above optimization problem. Inspired by , Adaptive Moment Estimation (Adam) Kingma , et al. (2015)[7] which is a first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order

moments. Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. The algorithms leverages the power of adaptive learning rates methods to find individual learning rates for each parameter.
We initialize teacher's network weight $T^*(A)$ using adaptive estimation update of T with respect to $L\left(T, A, D_t^{(\mathrm{tr})}\right)$. Then utilize optimal T' of $T^*(A)$ into $L\left(S, D_s^{val}, T^*(A)\right) + \lambda L\left(D_t^{val}, D_s^{val}, T^*(A), S\right)$ and obtain an objective O_s. Then we approximate $S^*\left(T^*(A)\right)$ using adaptive estimation update of S with respect to O_s. Finally, we plug T' and the approximation S_t of $S^*\left(T^*(A)\right)$ into $T^{**}(A) = T^*(A) - \alpha(T^*(A) - S^*\left(T^*(A)\right))$.

Similar to Kingma , et al. (2015)[7], as in this problem, we are interested in minimizing the expected value of this function, $\mathbb{E}[f(\theta)]$ w.r.t. its parameters θ. With $f_1(\theta), \ldots, , f_T(\theta)$ we denote the realisations of the stochastic function at subsequent timesteps $1, \ldots, T$. With $g_t = \nabla_\theta f_t(\theta)$ we denote the gradient, i.e. the vector of partial derivatives of f_t, w.r.t θ evaluated at timestep t. First of all, we find the optimal solution $T^{'}$of equation 1 $T^*(A)$ using update rule

$$T_t \leftarrow T_{t-1} - \alpha_T \cdot \widehat{m}_t / \left(\sqrt{\widehat{v}_t} + \epsilon\right) \tag{4}$$

where α is the step size, and the algorithm updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t) where the hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages. The moving averages themselves are estimates of the 1^{st} moment (the mean) and the 2^{nd} raw moment (the uncentered variance) of the gradient. $\widehat{m}_t$ and $\widehat{v}_t$. are bias-corrected estimates.
Secondly, we plugging T' into approximated objective $O_s = L\left(S, D_s^{val}, T^{'}\right) + \lambda L\left(D_t^{val}, D_s^{val}, T^{'}, S\right)$. Then we approximate $S^*\left(T^*(A)\right)$ using adaptive estimates update of S with respect to O_s :

$$S_t \leftarrow S_{t-1} - \alpha_S \cdot \widehat{m}_t / \left(\sqrt{\widehat{v}_t} + \epsilon\right) \tag{5}$$

Finally, we plug T_t and S_t into $T^{**}(A) = T^*(A) - \alpha(T^*(A) - S^*\left(T^*(A)\right))$. So we can update the teacher's network weight by

$$T'' \leftarrow T' - \eta\left(T^{'} - S_t\right) \tag{6}$$

Algorithm 1: Optimization Algorithm

initializations: α : Stepsize ;
$\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates ;
$f(T, S)$: Stochastic objective function with parameters, equations 1 and 2;
T_0,S_0 : Initial parameter vector ;
$m_0 \leftarrow 0$ (Initialize 1^{st} moment vector) ;
$v_0 \leftarrow 0$ (Initialize 2^{nd} moment vector) $t \leftarrow 0$ (Initialize timestep);
while T_t *Not Converge* **do**
  $t \leftarrow t + 1$;
  $g_t \leftarrow \nabla_T f_t\left(T_{t-1}\right)$ (Get gradients w.r.t. stochastic objective at timestep t) ;
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate) ;
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment ;
  estimate) $\widehat{m}_t \leftarrow m_t / \left(1 - \beta_1^t\right)$ (Compute bias-corrected first moment ;
  estimate) $\widehat{v}_t \leftarrow v_t / \left(1 - \beta_2^t\right)$ (Compute bias-corrected second raw ;
  Update teacher's network weight T_t using equation 4;
end
while S_t *Not Converge* **do**
  Update parameters $t, g_t, m_t, v_t, \widehat{m}_t, \widehat{v}_t$ similar to above;
  Update student's network weight S_t using equation 5;
  Update teacher's network weight T'' using equation 6;
end

4 Experiment

In this section, we will discuss the dataset, setup and analysis of experiment result on proposed method.

4.1 Datasets

To evaluate our proposed method, we choose the sentiment analysis as downstream task in text classification area as sentiment analysis is one of the most important technique in natural language process. In this project, we conducted experiments on sentiment analysis dataset Stanford Sentiment Treebank (SST). The SST dataset is one of GLUE Wang, et al. (2019) [8] benchmark dataset, the General Language Understanding Evaluation benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems. Introduced by Socher, et al. (2013) [9], SST dataset is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser (Klein and Manning, 2003) and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges. This new dataset allows us to analyze the intricacies of sentiment and to capture complex linguistic phenomena. For this experiment, the training, validation and test dataset split was same as GLUE with training dataset has 67,349 sentences or examples, validation dataset has 872 examples, and test dataset contains 1,821 examples. During teacher model training stage of teacher model, the training dataset is used as D_t^{tr} and the validation set is used as D_t^{val}. During student model training and fine tuning stage, the training and validation dataset are used as D_s^{tr} and D_s^{val} respectively.

Same as Devlin, et al. (2018)[10] , pre-training data for BERT model follows the existing literature on language model pre-training. For the pre-training corpus BERT use the BooksCorpus (800M words) and English Wikipedia (2,500M words).

4.2 Experiment Settings

We set up both student and teacher to have model size official $BERT_{BASE}$ (L=12, H=768, A=12, Total Parameters=110M) Devlin, et al. (2018)[10] where number of layers were denoted as L, the hidden size as H, and the number of self-attention heads as A. The input and output representation were also same as official BERT: the first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. Sentence pairs are packed together into a single sequence. Sentences were differentiated by special token ([SEP]).

The hyper-parameters setting were similar to Self-Distillation BERT Xu, et al. (2015)[6]. Student and teacher models' network weights were updated using AdamW optimizer. In order to reduce the primacy effect of the early training examples, we used AdamW optimizer with the warm-up proportion of 0.1 which means 10% proportion of training to perform linear learning rate warm-up. Epsilon for Adam optimizer is $1e^{-8}$. Base learning rate for BERT encoder is $2e^{-5}$, base learning rate for softmax layer is $1e^{-3}$, dropout probability of 0.1. For sequences of more than 512 tokens, we truncated them and choose head 512 as model input. For $BERT_{BASE}$, we initialize the batch size as 16, 6 epochs, and 1 gradient accumulation step . We run $BERT_{BASE}$ with random seed and save the checkpoints at every 500 steps.

4.3 Model Selection

The main hyperparameters in the fine-tuning stage we need to determine are the trade off parameter λ (which will be noted as LBT weight below) in minimizing the two parts loss function, and the coefficient α to update teacher model's network weights.
LBT weight We first evaluate our methods on SST dataset to investigate the effect of LBT weight in [0.3-0.5] and [0.6-0.8] respectively. The observations are when parameter is in range [0.3-0.5], the model has better results. Then evaluate the method with parameters in [0.3,0.35,0.4,0.45,0.5] respectively, and parameter = 0.45 has better result.
Coefficient α Similar to Xu, et al. (2015)[6], we set up the coefficient to update network weight as a

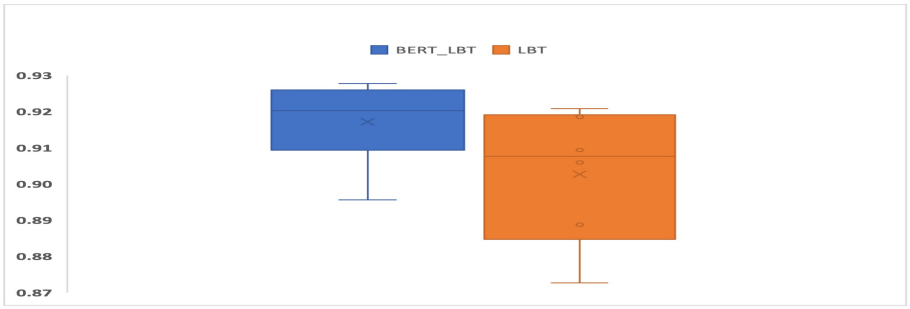

Figure 2: Boxplot for student model's training stability. Only 8000 labeled example in training set are used for training at each checkpoint.

Table 2: LBT-BERT has median of 92.9% accuracy in 5 runs and $\pm 0.16\%$ variation. Comparison on the dev sets of SST-2 benchmark results are reported by authors.

Model	Accuracy(%)	Param(M)
BERTPKD	89.4	52.2M
MobileBERT	91.7	15.1M
DistilBERT	91.3	52.2M
BERT-base	92.7	110M
LBT-BERT	92.9 ± 0.16	110M

function of

$$1 - (1 + \text{global step})/(10 + \text{global step})$$

Epoch and Batch Size The model results achieve better result when choosing 6 epochs and the performances stay relatively flat after 6. For batch size, we choose 32 per GPU and we fine-tune all models on 4 RTX 1070Ti GPU.

4.4 Model Result Analysis

Training Stability Inspired by Xu, et al. (2015)[6], we test the training stability conducting experiments to explore the effect of data order on the models. Generally, distinct random seeds can lead to substantially different results when fine-tuning BERT even with the same hyperparameters. In Xu, et al. (2015)[6] study, experiments are conducted with a set of data order seeds. One data order can be regarded as one sample from the set of permutations of the training data. In our study, we evaluate the training stability by comparing accuracy of student model's accuracy at different checkpoints when applying Learning by Teaching framework and without applying the framework. The accuracy of each checkpoint is evaluated by the same validation dataset 872 examples, and each training dataset has 8000 examples with random seed so the comparison of accuracy is evaluated on the same data.

Results of the boxplot in figure 2 shows that applying Learning by Teaching framework on $BERT_{BASE}$ model has higher accuracy and smaller variance than the single BERT fine-tuning across different checkpoints. This proves that the fine-tuned BERT with the LBT framework is less sensitive to the data order or dataset choice during training.

We compare the proposed method with several state-of-the-art knowledge distillation baselines including BERTPKD Sun, et al. (2019)[11], DistilBERT Sanh, et al. (2019)[3] and MobileBERT Sun, et al. (2020)[12]. In addition, we compared with original BERT Devlin, et al. (2019)[10]. Table 2 shows the classification accuracy(%), number of weight parameters (millions). From this table, we can observe that when our newly proposed algorithm is applied to SST-2 dataset, the accuracy rate is improved from 92.7% to 92.9% compared to original $BERT_{BASE}$ and performed better than other knowledge distillation BERT models above. This demonstrates the effectiveness of our method in fine tuning BERT model to better network weights. To further analyze the convergence of the proposed new method, we plot converge curve while training. The training curves on SST-2 dataset are shown in Figure 3. With LBT framework, the BERT model get relatively better improvement at

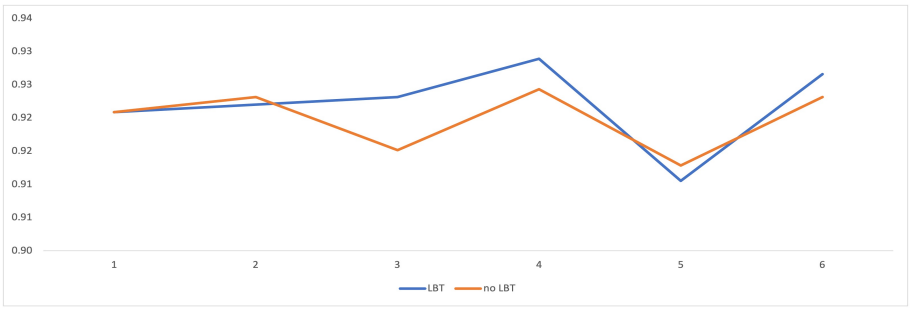

Figure 3: Test accuracy rates(%) during different epoch

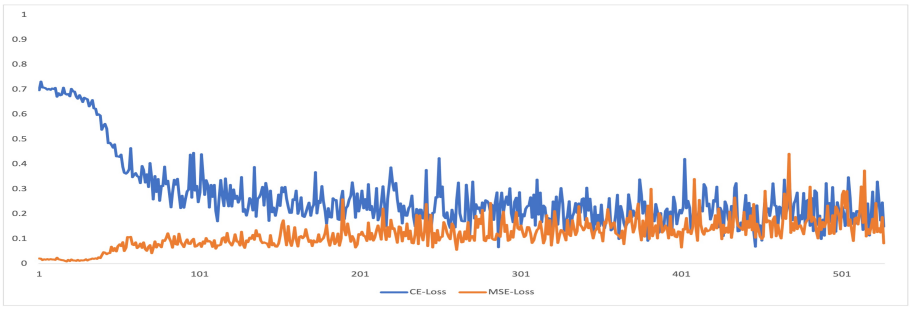

Figure 4: Loss curve of student model

epoch 3, 4 and 6. Following method in Xu, et al. (2015)[6], to further analyze the reason for this observation, we also record the loss curve of cross-entropy (CE) loss and mean-squared-error (MSE) loss, as shown in Figure 4. The observations were similar to previous study: when training begins, the CE loss dominates the optimization objective. In the last phase of training, the cross-entropy loss becomes small, and a large proportion of gain also comes from LBT framework. Therefore, although optimizing the CE loss at the end of the training phase cannot continue improving the performance of BERT, applying LBT framework will continuously enhance the generalization and robustness of BERT model through teacher.

5 Conclusion and Future Work

In this project, we propose a new strategy to improve BERT model performance by applying learning by teaching (LBT) inspired by Xie, et al. (2020)[4]. In LBT BERT, a teacher BERT model improves its learning outcome on text classification task by teaching a student model to perform well on this task. Intuitionally, teacher model and student model can benefit from each other. The learning outcome is improved by making the teacher model as an “feedback receiver” of the student model within the student model’s fine-tuning stage. Intuitionally, the teacher learned and transfer the knowledge (network weight) to the student. The student will validate itself and provide “feedback” to the teacher for further improvement, where the "feedback" is a combination of CE loss and MSE loss. We applied newly proposed algorithm on benchmark dataset SST-2 and the results demonstrate the effectiveness of our method on sentiment analysis task in text classification by improve model accuracy to 92.9% and relatively less sensitive to data order.
In future, we will conduct evaluate our method on GLUE benchmark dataset other than sentiment analysis downstream tasks to demonstrate the effectiveness of our method on natural language processing. Furthermore, we will investigate a better fine-tuning strategy by integrating our newly proposed method into an optimization algorithm.

References

[1] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, Donald E. Brown. (2019) Text Classification Algorithms: A Survey. $arXiv : 1904.08067$

[2] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. $arXiv : 1906.08237$

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. $arXiv : 1910.01108$

[4] Sheth, Parth, Xie, Pengtao (2020) Learning by Teaching, with Application to Neural Architecture Search. $arXiv : 1910.01108$

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. (2015a) Distilling the knowledge in a neural network. $arXiv : 1503.02531$

[6]Yige Xu, Xipeng Qiu, Ligao Zhou, Xuanjing Huang. (2020) Improving BERT Fine-Tuning via Self-Ensemble and Self-Distillation. $arXiv : 2002.10345$

[7]Diederik P. Kingma, Jimmy Lei Ba. (2015) Adam: Method for Stochastic Optimization. In ICLR 2015

[8]Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman1. (2019) Glue: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In ICLR 2019

[9]Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts. (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In EMNLP 2013

[10]Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. $arXiv : 1810.04805$

[11]Siqi Sun, Yu Cheng, Zhe Gan, Jingjing Liu. (2019) Patient Knowledge Distillation for BERT Model Compression. $arXiv : 1908.19355$

[12]Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, Denny Zhou. (2020) MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. $arXiv : 2004.02984$