Assignment 3 Report

Name: Hongzhao Tan

MacID: Tanh10

Student #: 400136957

1. Introduction and Theory

TANDA is a technique that performs two-step finetuning on pretrained Transformer-based model(s) for natural language classification tasks. The two-step finetuning includes, (i) a transfer step that finetunes the pretrained model into a model for Answer- Sentence Selection (AS2) task using a large and high-quality dataset called ANSQ; (ii) an adapt step that continue finetunes the transferred model into a model for a specific target domain.

The TANDA technique has made finetuning easier, more stable and more robust to noisy data, and achieved significantly higher accuracy than doing traditional one-step finetuning. The stability introduced by TANDA offers low variance of model accuracy between 2 consecutive training epochs, and between models that have similar accuracy on development/validation set. TANDA also enables modularity and efficiency, meaning that after a pretrained Transformer-based model(s) being transferred using the large ANSQ dataset, the model only needs a much smaller dataset in the adapt step to be finetuned for a specific target domain, so that fixed a critical issue which it is usually very hard to find enough data of a specific domain to make a pretrained model to effectively work on the target domain by performing traditional one-step finetuning.

The architecture of Transformer model benefits the TANDA technique in two ways: (i) by pretraining a Transformer model, the model will learn to generalize the input sequence while providing the same output sequence, which can usually in turn improve the performance of the model after applying TANDA on it; (ii) The Transformer architecture made the Transformer-based model can still remember the knowledge/information that is already encoded in it when it is being finetuned, so that after the two-step finetuning the model will adapt to a target domain, while also keeping the knowledge it has learned/encoded from pre-training.

2. Preparation and Dataset Understanding

The ANSQ dataset has been used for the transfer learning step, and the TREC-QA dataset has been used for the adaptation step for domain-specific finetuning. The ANSQ dataset has columns namely “question”, “sentence” and “label”, while the TREC-QA dataset contains “question”, “answer”, and “label” columns. The values in “label” columns in both datasets are either 0 or 1, indicating whether the string values in the “answer” / “sentence” column are the true answer (1) of the “question” values on the same row or not (0). Then ANSQ dataset has been split into training set and validation set, while the TREC-QA dataset has been split into training, validation and test sets.

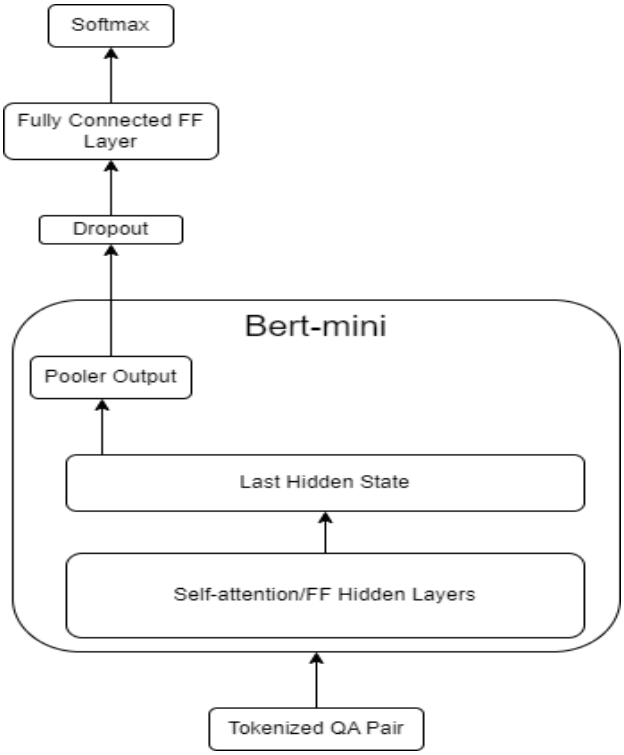
There are 3 main challenges that have been found during the EDA:

1. The size of the ANSQ dataset was too large to use the whole dataset to train/finetune the model, given the limit of time.
2. The number of data samples for each of the unique label values (classes) were very imbalanced. Adjustment on the loss function that will be used for training the models will be needed.
3. Some of the QA pairs in the ANSQ dataset was too long in length, and their sequences of word embeddings could exceed the input length limit for the Transformer-based model (e.g., for Bert-base-uncased, the limit of input length is 512 tokens).

Details of EDA can be found in the attached Jupyter Notebook.

3. Model Implementation

Bert-mini[1], a smaller variant of pre-trained Bert-base model, has been employed as the base of the Transformer-based model that was built in this assignment's work for QA pair classification task. Same as Bert-base model, Bert-mini has 512 input features and 0.1 of hidden state dropout probability, while it only has 256 hidden-state dimensions and 4 hidden layers. The same tokenizer that was used to pretrain Bert-mini was used to tokenize the QA pairs from the ANSQ and TREC-QA datasets and turn them to numerical sequences (i.e. sequences of word embeddings). During tokenization, every QA pair was turn into form of [CLS][tokens of question][SEP][tokens of answer][SEP]. To make the model be able to work on QA task, a classification head has been added upon the pooler output of the Bert-mini (i.e. the last-layer hidden state of the leading [CLS] token further transformed by a linear layer and a Tanh activation function). The classification head includes a dropout layer with the same dropout rate as Bert-mini, a fully connected linear layer with the number of input features equals to the size of hidden state of Bert-mini (i.e. 256) and the number of output features equals to the number of unique label values (classes) (i.e. 2), and a Softmax function that computes the probability distribution of the QA pair(s) across 2 classes. It can be visualized as below:



Details of model implementation can be found in the QATransformer class in the attached Jupyter Notebook.

Three sets of Transformer-based QA models have been initialized using the QATransformer class:

(1) A set of QA models with only the newly added classification head finetuned (or rather pretrained, as the parameters in the classification head have never been trained before) using a dataset merged from the ANSQ and TREC-QA training datasets in one step. The model in this set with the best performance on a merged validation set from ANSQ and TREC-QA validation sets was used as the baseline model in later comparison.

(2) A set of QA models with traditional one-step finetuning performed on all the parameters of the whole model (i.e. both the new classification head and the pretrained Bert-mini), using a dataset merged from the ANSQ and TREC-QA training datasets in one step. The model in this set with the best performance on a merged validation set from ANSQ and TREC-QA validation sets was used in later comparison.

(3) A set of QA models with TANDA two-step finetuning performed on all the parameters of the whole model (i.e. both the new classification head and the pretrained Bert-mini), using the ANSQ training/validation set for training/validating the models in the transfer step, and the TREC-QA training/dev set for training/validating the models in the adapt step. The model in this set with the best performance on the TREC-QA dev set was used in later comparison.

Each model in each of the 3 sets have been trained/finetuned with different learning rate values.

Details of training/finetuning implementation can be found in the attached Jupyter Notebook. More details of the training process will be discussed in the next section.

4. Experimental Setup and Evaluation

For both the first two sets of the models that have been mentioned in the end of Section 3, each of the models has been trained/finetuned with one of the learning rate values among 1e-6, 2e-6, 5e-6, 1e-5, and 2e-5, through 9 epochs. For the set of model that have been finetuned with the TANDA technique, each of the models has been finetuned through a transfer step with learning rate value being either 1e-6 or 2e-5, through 9 epochs, and then an adapt step with one of the learning rate values among 1e-6, 2e-6, 5e-6, 1e-5, and 2e-5, through 18 epochs. The selections of candidate learning rate values for the transfer and adapt steps was the same as the candidate values that have been mentioned in the TANDA research paper (Garg, Vu and Moschitti) [2].

All the models have been trained/finetuned using Adam as the optimizer, cross entropy loss as the loss function, and 128 as the batch size of data loader. As mentioned in Section 2, the numbers of data samples for each of the 2 classes/labels were highly imbalanced. Hence, different weights/scales have been assigned to the cross entropy loss function for misclassifying data samples of different classes. The weights were computed according to the ratio between the number of lable-1 data samples and the number of label-0 data samples in the training set of ANSQ/TREC-QA dataset.

For each model, after each epoch of training/finetuning, the tranined model's performance (i.e. accuracy and F1-score) has been validated on the training and validation sets of the dataset the model was being trained with. The line plots of accuracies and F1-scores, for each of the 3 sets of models on the training and validation sets across each epoch of finetuning, where each model's performance corresponds to a line in the plots with a legend named with the learning rate the model was finetuned with, are demonstrated as below:

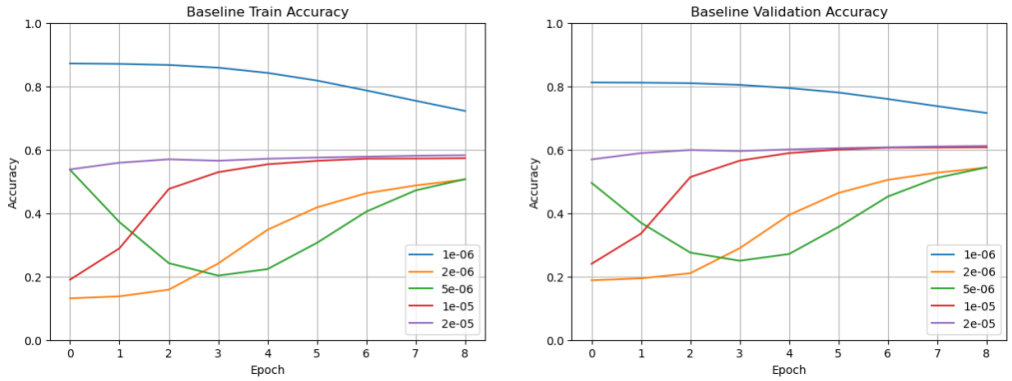


Fig. 1 Baseline Models' Training and Validation Accuracies Through Epochs

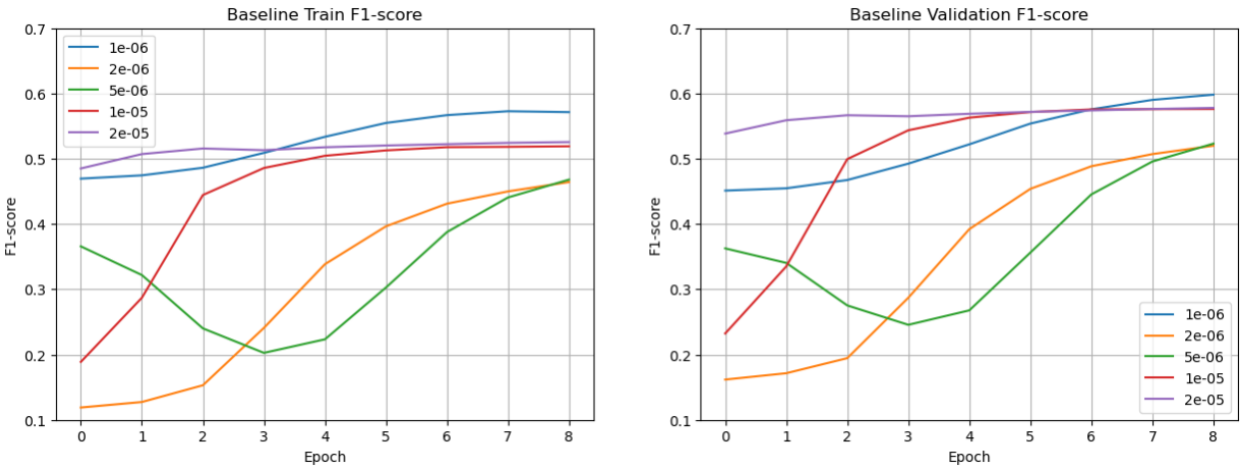


Fig. 2 Baseline Models' Training and Validation F1-scores Through Epochs

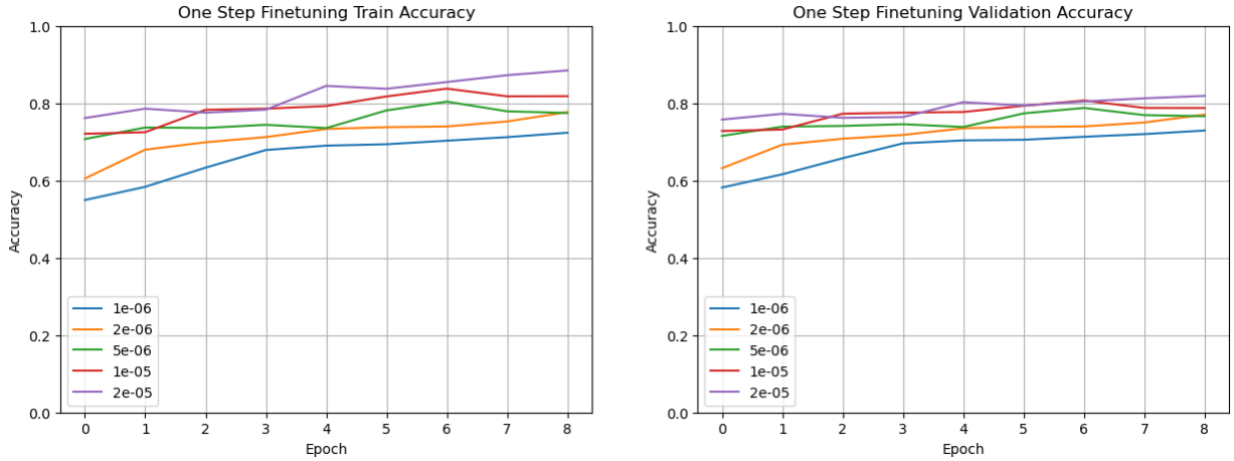


Fig. 3 One-step Finetuned Models' Training and Validation Accuracies Through Epochs

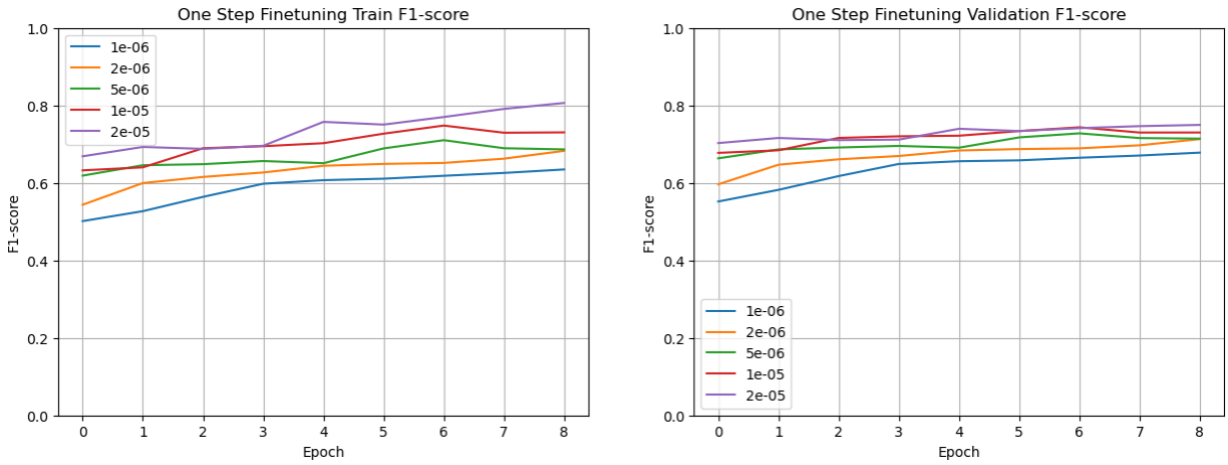


Fig. 4 One-step Finetuned Models' Training and Validation F1-scores Through Epochs

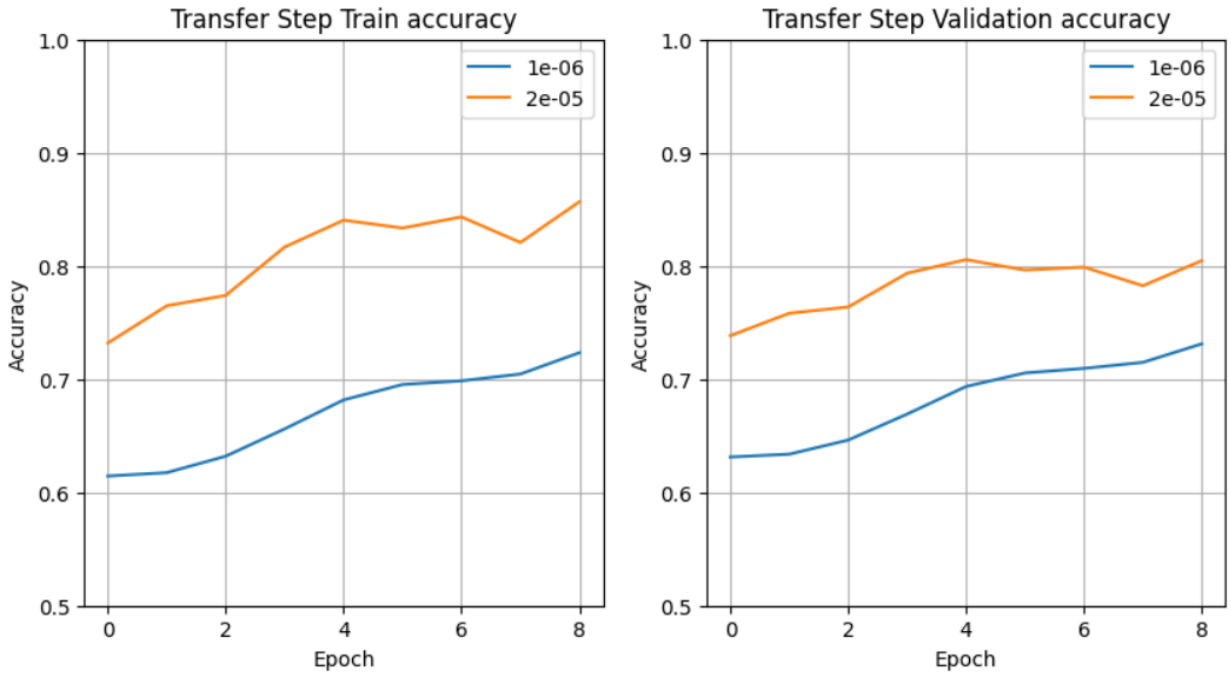


Fig. 5 Training and Validation Accuracies of Models Finetuned using TANDA Technique During the Transfer Step Through Epochs

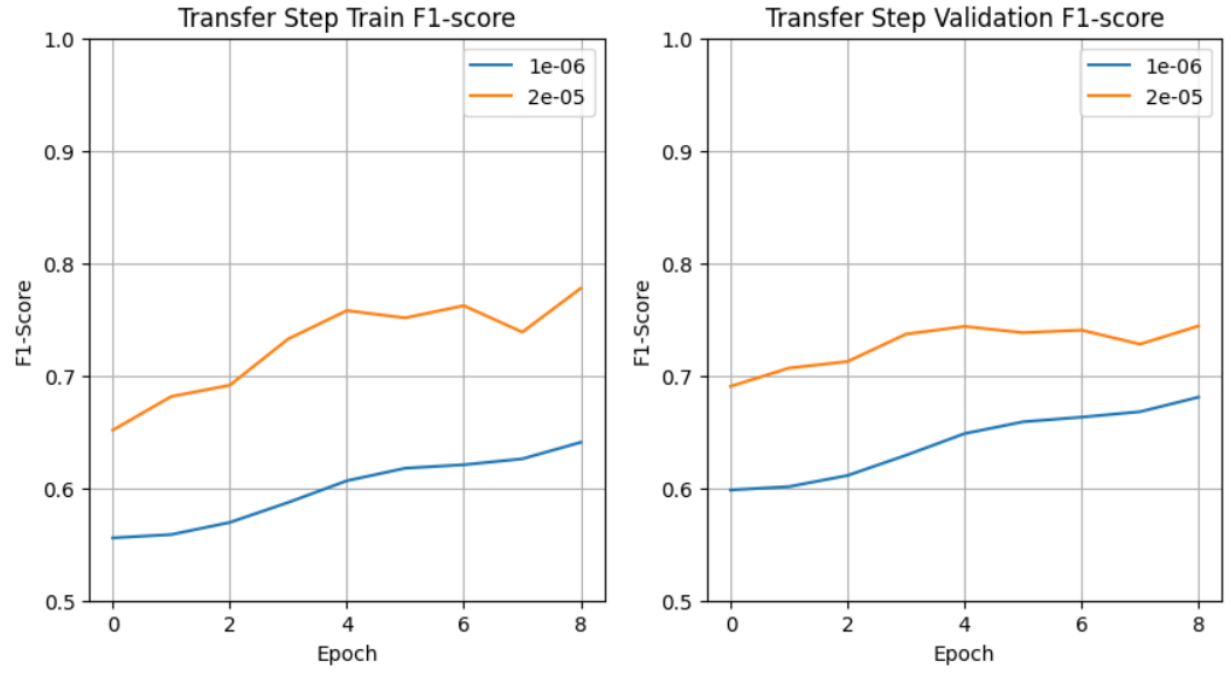


Fig. 6 Training and Validation F1-scores of Models Finetuned using TANDA Technique During the Transfer Step Through Epochs

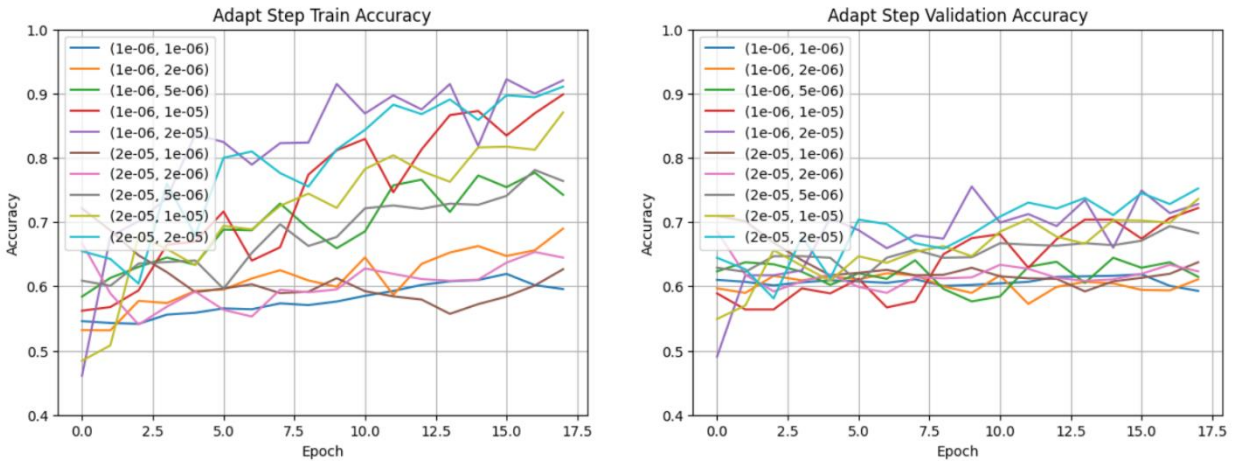


Fig. 7 Training and Validation Accuracies of Models Finetuned using TANDA Technique During the Adapt Step Through Epochs

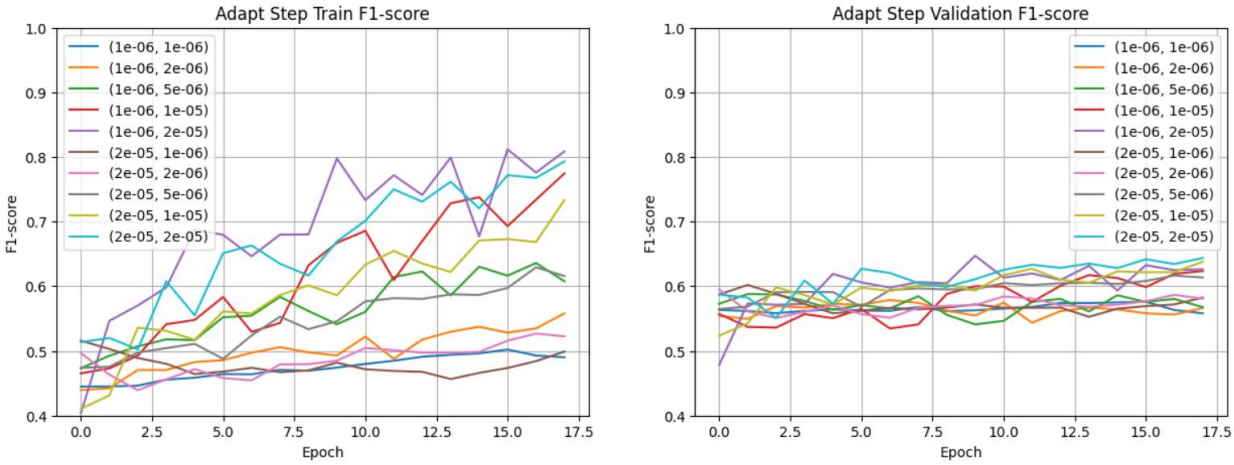


Fig. 8 Training and Validation F1-scores of Models Finetuned using TANDA Technique During the Adapt Step Through Epochs

From Fig. 1 and Fig. 2, it can be found that both the highest accuracies and F1-scores have been achieved by the baseline model that has been trained with learning rate of 1e-6. Interestingly the accuracies of the model have been decreasing while the F1-scores have been increasing as the number of epochs increased. This could because that, during the first a few epochs, the model has shown an on-off behavior where it classified all QA pairs to be labelled as 0. Because the dataset was highly imbalanced, the on-off behavior would lead to high accuracy while low F1-score, since the precision ($\frac{TP}{TP+FP}$) of the model would be 0. As the model keep learning from the differently weighted cross entropy loss through epochs, the model would classify some QA pairs to be label-1, which include some QA pairs that were actually label-0, so in turn increasing its F1-score while decreasing its accuracy.

According to Fig. 3 and Fig. 4, among the models that have been finetuned using traditional one-step finetuning, the model that has been finetuned with learning rate of 2e-5 has achieved the highest accuracy and F1-score on both the training and validation sets, after 9 epochs.

Based on Fig. 5 and 6, for the transfer step, the training and validation accuracy and F1-score achieved by using 2e-5 as the learning rate, have been higher than those by using 1e-6 as the learning rate, across all 9 epochs. However, for the adapt step, according to Fig. 7 and 8, the highest training accuracy, validation accuracy, validation F1-score and nearly the highest validation accuracy have been achieved by the model that was using 1e-6 as the learning rate in the transfer step and 2e-5 as the learning rate in the adapt step, after the model has been finetuned through 10 epochs (note that the plots are 0-indexed) in the adapt step. Interestingly, the optimal learning rates that were found in this assignment's work, for the transfer and adapt steps, aligned with the optimal learning rates that have been mentioned in the TANDA research paper (Garg, Vu and Moschitti) [2].

As the last step, to evaluate, the 3 optimal models that have found from the 3 sets of models (i.e. the baseline model that was pretrained only on the classification head with 1e-6 learning rate through 9 epochs; the model that was one-step finetuned with 2e-5 learning rate through 9 epochs; the model that was performed TANDA two-step finetuning on, with 1e-6 learning rate in the transfer step through 9 epochs, followed with 2e-5 learning rate in the adapt step through 10 epochs), have performed QA pairs classification task on the test set of TREC-QA dataset.

The **optimal baseline model** has achieved **0.514 accuracy and 0.435 F1-score**. The **optimal one-step finetuned model** has achieved **0.752 accuracy and 0.597 F1-score**. The **optimal model that has been performed TANDA technique on**, has achieved **0.774 accuracy and 0.635 F1-score**.

5. Discussion and Conclusion

During the assignment, a few challenges have taken place. To address the first challenge that have been found during EDA and mentioned in Section 2, from the ANSQ dataset, only 50,000 label-0 sample QA pairs in the training set, and 18228 label-0 sample QA pairs in the validation set were used in this assignment. It would not only allow the assignment to be done within the given time limit, but also can increase the ratio of label-1 samples in the whole set of samples that was used in this assignment, so that information of the label-1 samples can be less diluted and better learned by the models. To deal with the second challenge found from the EDA, different weights have been assigned to the loss function misclassifying QA pairs of different label value, as discussed in Section 4. For the last challenge found from the EDA, the QA pairs whose lengths of their corresponding sequence of word embeddings were exceeding the input length of the model (i.e. 512), were simply discarded, as there was already a lot more than enough sample data left after removing them. Another challenge has been encountered

when figuring out how to transform the outputs of the base pre-trained Bert-mini model (i.e. the last hidden states and the pooler output) to a form that can be used for QA pair classification. To tackle this, a classification head has been added upon the pooler output of the Bert-mini model as a modification on the model architecture, details about the classification head have discussed in Section 3.

As proven by the selected models' accuracies and F1-scores on the test set of TREC-QA, mentioned in the end of Section 4, the TANDA two-step finetuning technique can greatly improve the performance of pre-trained baseline model on AS2 task of a specific domain. The technique can also let the model achieve better test results than performing traditional one-step finetuning when using the same training data. Nevertheless, there could still be some areas for improvement for the TANDA technique. As it has been proven that TANDA two-step finetuning could let the model perform better than traditional one-step finuetuning, then N-step finetuning for N > 2, might be able to improve model's performance even further on AS2 task of more specific domain(s). For instance, a pretrained-model could be first transferred with the ANSQ dataset, then adapt to a dataset of QA pairs about general medical consultation between patients and doctors, and at last further adapt to a dataset of QA pairs that are related to specific medical consultation about sickness on lung/head/stomach/etc..

The TANDA two-step finetuning technique could also potentially be applied on NLP tasks other than AS2. For example, similar to transforming the pooler output of pretrained Transformer-based model to classify QA pairs or simply a text sequence, where the pooler output was computed from the last hidden state of the leading [CLS] token, it might be feasible to compute the "pooler output" of the last token's last hidden state, then transform the "pooler output" to predict/generate the next word(s). If a large and high-quality dataset for next-token prediction can be given, the TANDA technique could be applied on a pretrained Transformer-based model for word generation tasks and the finetuned model could work as a generative language model for a specific target domain.

References

[1] "prajjwal1/bert-mini · Hugging Face," *huggingface.co*. https://huggingface.co/prajjwal1/bert-mini (accessed Mar. 13, 2024).

[2] S. Garg, T. Vu, and A. Moschitti, "TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection," *arXiv:1911.04118 [cs]*, Nov. 2019, Available: https://arxiv.org/abs/1911.04118