

Can We Teach the Model Twice?

NLP Final Project Report

Yongqian Peng

PKU Yuanpei College

2100017731@stu.pku.edu.com

Abstract

Recent work on solving Natural Language Processing (NLP) tasks by fine-tuning pre-trained language models (LMs) achieves great success in NLP applications. In realistic scenarios, fine-tuning an LM to fulfill a need is a common technique. However, when the need changes, we need to balance the cost and the performance for fine-tuning a model, considering fine-tuning the initial pre-trained model or the one that has been fine-tuned once. We designed experiments to compare both methods and analyzed the mechanisms through ablation studies. Experiments and ablation studies were conducted, which shows that fine-tuning the model twice is considerable under certain conditions. All the codes and data can be downloaded on <https://github.com/PPYYQQ/2023Fall-NLP-Final-Project/tree/main>.

1 Introduction

In recent years, the field of Natural Language Processing (NLP) has witnessed remarkable advancements, primarily driven by the successful utilization of fine-tuned pre-trained language models (LMs). This approach has demonstrated exceptional efficacy across various NLP tasks, becoming a cornerstone in the development of state-of-the-art applications. As practical NLP scenarios often demand adaptability, fine-tuning pre-trained models has become a common technique to make them fulfill specific needs.

However, the dynamic nature of real-world requirements introduces a crucial consideration in the fine-tuning process—the balance between cost and performance. When the demand evolves, decision-makers face the dilemma of whether to fine-tune the initial pre-trained model or the one that has been through one round of fine-tuning. This decision is important, influencing both the resource investment and the resulting model performance.

In response to this challenge, our research delves into a tentative exploration of the fine-tuning land-

scape, aiming to provide insights into the most effective strategies under varying circumstances. We present designed experiments that systematically compare the outcomes of fine-tuning the initial pre-trained model against the model that has been fine-tuned once. Through analyzing the ablation studies, we seek to discover the hidden mechanisms, trying to find the conditions under which either approach proves more advantageous.

The core focus of our investigation lies in addressing the critical question: Is it more pragmatic to fine-tune a language model twice, and if so, under what specific conditions does this strategy yield significant benefits? To this end, we employ ablation studies to dissect the contributing factors to the observed performance differences.

This research not only contributes to the optimization of fine-tuning strategies but also advances our understanding of the intricacies involved in adapting language models to evolving NLP requirements. By elucidating the trade-offs and advantages associated with different fine-tuning approaches, our findings aim to inform practitioners and researchers alike, fostering a more informed decision-making process in the ever-evolving landscape of natural language processing.

2 Experiment

2.1 Datasets and Baseline

Datasets: We use SemEval2021 task 7 (Meaney et al., 2021), SemEval2014 task 4 (Pontiki et al., 2014) and a dataset we built with the data from Reddit following previous methods of Tang et al., 2022. (Table 1)

Baseline We use Deberta-v3-base (Pengcheng He, 2020) on HuggingFace to directly fine-tune the model.

2.2 Experiment Settings

We define two tasks with the datasets we used. TaskA is developed with SemEval2021 task 7

Task	Dataset	NOL	SOTr	SOTe
TaskA	SemEval2021	2	8.0k	10.0k
TaskB	Reddit Data	3	5.3k	1.3k

Table 1: Statistics of Datasets. NOL means the number of label. SOTr means the size of the training set. SOTe means the size of the test set.

(Meaney et al., 2021), which is a binary classification task, discriminating whether a sentence humorous or not humorous. TaskB is developed with the dataset we built with Reddit, which is a classification task with three categories of labels. With these two datasets, we consider two methods of fine-tuning the model to solve TaskB: 1)directly fine-tuning the initial pre-trained model on TaskB 2)or fine-tuning the initial pre-trained model on TaskA and then fine-tuning it on TaskB. To study the balance between cost and performance under various conditions, we consider different training steps in the fine-tuning process. For the fine-tuning twice paradigm, the first round of training leads to an excellent performance of the models in solving TaskA with an accuracy higher than 95% .

2.3 Implementation Details

Architecture: We adopt Deberta-v3-base as our backbone LM. We adopted the approach of fine-tuning the entire model and in the fine-tuning twice setting, we saved the parameters of the entire model and used them in the second round of fine-tuning.

Hyperparameters: Unless otherwise stated, the same hyper-parameters are used in all experiments. We considered fine-tuning the model for 500, 1000, 2000 and 3000 steps respectively and other hyper-parameters fixed. For a certain number of steps, we took 5 rounds of fine-tuning with random seeds into consideration to reduce the impact of randomness. The learning rate is set to 3e-6. The weight decay is set to 0.1. The batch size is set to 32. Besides, the Trainer package of HuggingFace is used to fine-tune the model.

2.4 Evaluation Results and Analysis

As we can see from Figure 1, in the settings of 500 and 1000 training steps, the models are not completely trained, while in those of 2000 and 3000 training steps, the models converge well, which shows two patterns of training representing the conditions of lack and sufficiency in computational resources respectively.

Method	500s	1000s	2000s	3000s
Twice	0.667	0.733	0.749	0.757
Directly	0.606	0.717	0.753	0.761

Table 2: The accuracy of models with different steps of fine-tuning and methods.

2.4.1 Stability

Firstly, the lines show the median over seeds, while the shaded areas stand for the mean \pm standard deviation over five different random seeds. The shaded areas show that under all conditions, the training processes are more stable with the method of fine-tuning twice. The fewer the training steps are, the more improvement in stability is observed. A possible explanation for the differences in stability is that in the first round of training, the model converges to solve a specific task. Compared to the initial pre-trained model which was trained with various pre-training tasks, the model already fine-tuned once learns a more specific distribution, which makes the second round of fine-tuning more stable.

2.4.2 Performance

Secondly, from Figure 1 and Table 2, the resulting performances show two patterns: 1) When the computational resources are limited, fine-tuning the model twice leads to higher resulting performances. Besides, The fewer the training steps are, the more improvement in performance is observed. 2) When the computational resources are sufficient, no obvious difference in resulting performances can be observed. Moreover, the method of fine-tuning the model twice hurts the resulting performance.

Possible explanations for the phenomena are that: 1) the model learns the semantics of humor and the task type of classification, which leads to training cost savings. 2) the general ability of the pre-trained model is hurt in the process of the first round of training, which leads to a decline in the resulting performances when the computational resources are sufficient.

2.4.3 conclusion

The method of fine-tuning the model twice is considerable, which can lead to a more stable training process and higher resulting performance when the computational resource is limited.

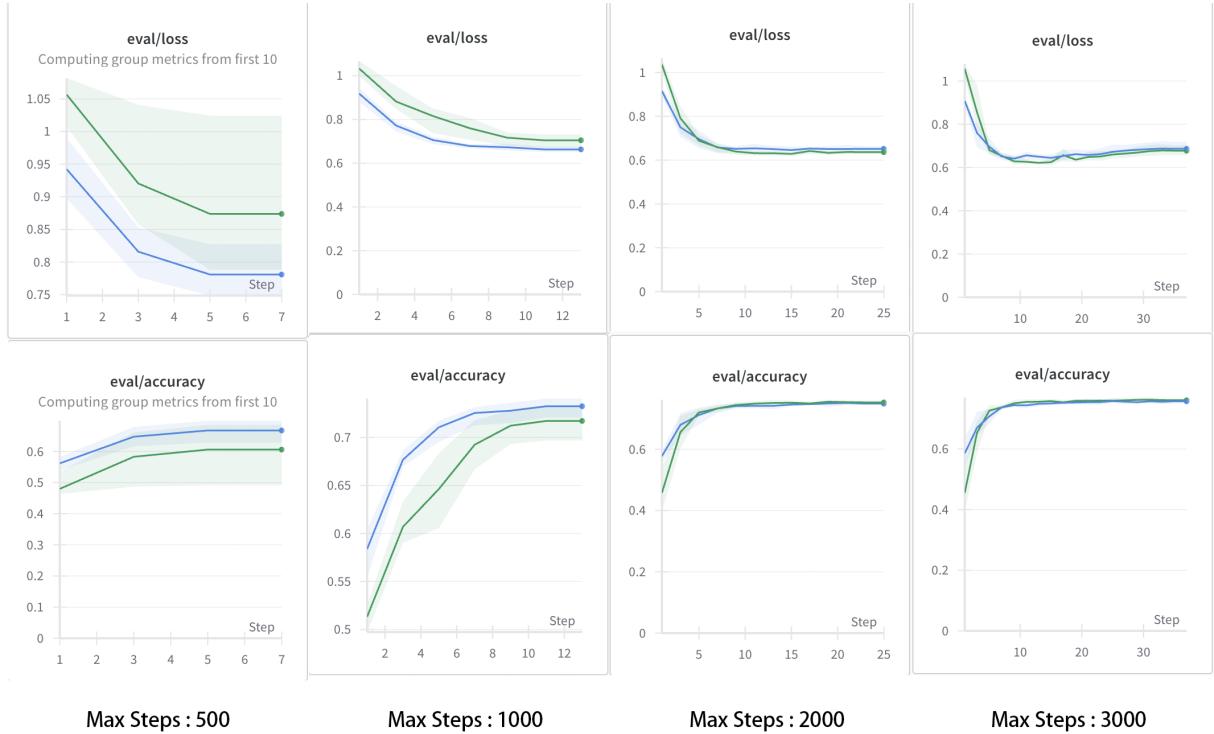


Figure 1: The evaluation loss and accuracy graphs of different steps of fine-tuning. The blue lines stand for results of fine-tuning twice while the green ones stand for results of fine-tuning directly

2.5 Ablation Study

To discover the contributing factors to the observed performance differences. We conducted another two experiments trying to figure out: 1) Does the exposure to the semantics of the data lead to the differences? 2) Does the experiences of doing the same type of task lead to the differences?

2.5.1 Datasets

Another two datasets are used. For *TaskB_1*, we use the SemEval2014 task 4 (Pontiki et al., 2014), which is a classification task with 3 categories of labels to discriminate the sentiment of comments for the restaurant. *TaskB_1* is the same type of task with *TaskA*, while the semantic is different. For *TaskB_2*, we use the SemEval2021 task 7 (Meaney et al., 2021), which is a regression task to rate the humorous sentences. *TaskB_2* is the different type of task with *TaskA*, while the semantic is the same. It is worth noting that, we split the test set of the SemEval2021 task 7 (Meaney et al., 2021), and extracted all the humorous sentences in the test set and the corresponding ratings of humor, which are originally in the dataset. (Table 3)

Task	Dataset	NOL	SOTr	SOTe
<i>TaskA</i>	S.E.2021	2	8.0k	10.0k
<i>TaskB_1</i>	S.E.2014	3	3.4k	1.2k
<i>TaskB_2</i>	S.E.2021	1	6.6k	3.4k

Table 3: Statistics of Datasets for ablation study. NOL means the number of label. SOTr means the size of the training set. SOTe means the size of the test set.

2.5.2 *TaskB_1*

TaskB_1 is the same type of task with *TaskA*, while the semantic is different. The result is shown in Figure 2, from which we can observe phenomena similar to those seen earlier. However, the semantic of humor is not helpful for the task, which indicates that the experiences of solving the same type of task are a contributing factor for the augmentation. Besides, the first round of training badly hurt the resulting performances for the second task when the computational resources are sufficient.

2.5.3 *TaskB_2*

TaskB_2 is the different type of task with *TaskA*, while the semantic is the same. It is worth noting that, the setting is slightly different for the Trainer package is not used and the learning rate is fixed through the training process. The result is shown in

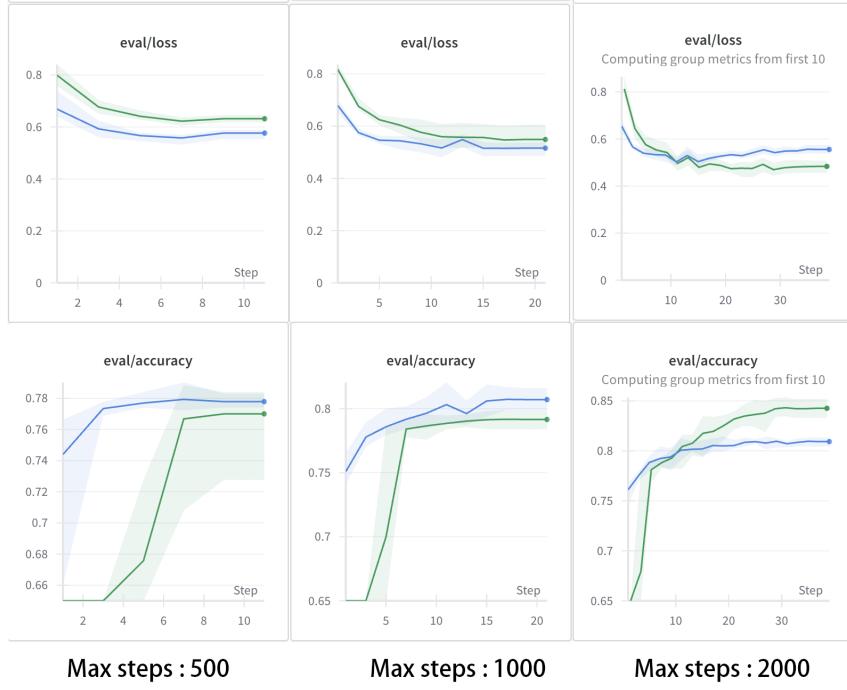


Figure 2: The evaluation loss and accuracy graphs of different steps of fine-tuning for *TaskB₁*. The blue lines stand for results of fine-tuning twice while the green ones stand for results of fine-tuning directly

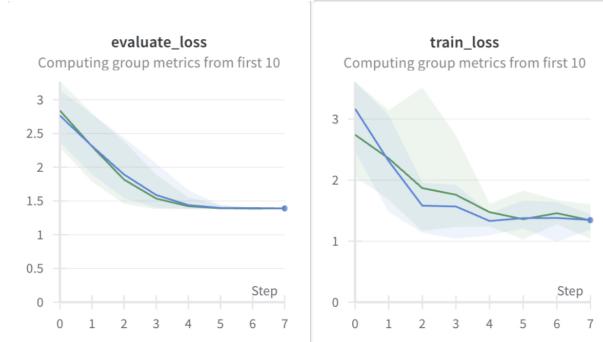


Figure 3: The evaluation loss and training loss graphs of fine-tuning processes for *TaskB₂*. The blue lines stand for results of fine-tuning twice while the green ones stand for results of fine-tuning directly

Figure 3, from which no obvious augmentation can be observed. The semantic of humor is almost the same for *TaskA* and *TaskB₂* under the settings but when the task type changes the method of fine-tuning twice is harmful, which indicates that the exposure to similar data does not help.

3 Conclusion

Through the experiments and the ablation study, we have come to a conclusion. The method of fine-tuning twice is considerable when 1) the task types of the first and the second rounds are similar or

the same and 2) the computational resources are limited. However, when the task types are different or the computational resources are sufficient, the method can not bring positive effects but negative ones.

4 Limitation and Future Work

In this project, the first round of the training is completed and the dataset is bigger than those in the second round. Experiments in different settings should be conducted in the future. We did not strictly make the datasets for the second round of training the same size, which could bring potential issues.

Acknowledgements

I would like to express my sincere attitude to Prof.Li and all three TAs Haowei Lin, David Zhu and Qinyu Chen who organized the NLP course with their effort.

References

- J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. *SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense*. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*,

pages 105–119, Online. Association for Computational Linguistics.

Jianfeng Gao Weizhu Chen Pengcheng He, Xiaodong Liu. 2020. **Deberta: Decoding-enhanced bert with disentangled attention**.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. **SemEval-2014 task 4: Aspect based sentiment analysis**. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Leonard Tang, lexander Cai, Steve Li, and Jason Wang. 2022. **The naughtyformer: A transformer understands offensive humor**.