Can We Teach a Model Twice?

Zhuoying Li
Yuanpei College
Peking University
joy@stu.pku.edu.cn

1 Introduction

In the industrial sector, to meet specific requirements, developers often fine-tune pre-trained models based on open source or previously trained models. This fine-tuning process adapts the pre-trained model to specific downstream tasks. When facing new requirements, developers encounter the following issue: should they use a previously fine-tuned model for old requirements or start training from scratch using a pre-trained model that has not undergone fine-tuning?

In this paper, we use three tasks to explore whether a model can be taught twice. Then we go a step further to explore whether it can learn various tasks simultaneously [1].

2 Tasks settings

2.1 Abstract to title (A2T)

In this task, the model's objective is to create a title derived from a provided abstract. For this purpose, we have utilized the cs.AI subset of the arXiv dataset [2]. This subset comprises approximately 40,000 entries. Of these, we have allocated 36,000 for the training set, 2,000 for the validation set, and the remaining 2,000 for the test set. The evaluation metrics for this task are ROUGE and sacreBLEU.

2.2 Translation

In this task, our model needs to translate English to Chinese. We employ the ALT dataset [4], which contains 20,000 samples. Of these, we allocate 18,000 for training, 1,000 for validation, and the remaining 1,000 are reserved for testing. The evaluation metric is sacreBLEU.

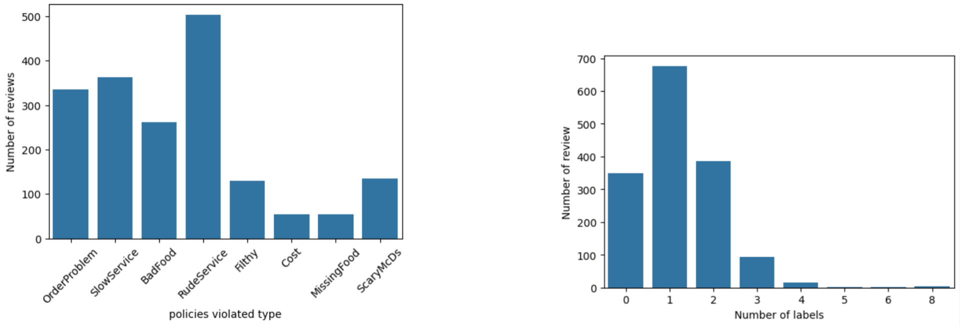
2.3 Restaurant review classification

This is a multi-label classification task that requires the model to classify the category of bad reviews of restaurant. We employ the McDonalds Review Sentiment dataset [2], which includes 1,525 entries. Of these, we use 1,143 for training, 191 for validation, and 191 for testing.

There are eight types of label in this dataset: *OrderProblem, BadFood, SlowService, Flthy, Rudeservice, Cost, ScaryMcDs, MissingFood.* Fig. 1 shows an analysis of this dataset. We can see that *Rudeservice* occurs most frequently among all the labels. Besides, the majority of the instances in this multi-label task are assigned less than three labels, and quite a few data points do not fall under any category (containing zero labels). The evaluation metrics are accuracy and F1 score. Notice that the term "accuracy" here refers to absolute accuracy, which means that it is only considered correct if every single label is classified accurately. In contrast, the F1 score represents the average of the F1 scores for each category.

[1] Github: https://github.com/joyli-x/NLP-Final-Project.git. **Because the machine of BIGAI is quite unstable and crashed a few times, I borrow a RTX 3090 from a research lab. Therefore the user who makes commitment is not me.**

[2] https://data.world/crowdflower/mcdonalds-review-sentiment



(a) Statistics of the number of occurrences of each label in the dataset.

(b) Distribution of the number of labels per sample in the dataset.

Figure 1: Analysis of McDonalds Review Sentiment dataset.

Finetuned Datasets	sacrebleu	rougeL	rouge1	rouge2
A2T	11.5420	0.3701	0.4067	0.2304
res -> A2T	**11.6576**	**0.3708**	**0.4120**	**0.2306**
trans -> A2T	11.1662	0.3675	0.4054	0.2263

Table 1: Results on A2T. "res", "trans" stand for "restaurant review classification" and "translation" respectively. And "res -> a2t" means "fine-tune on restaurant review classification dataset then fine-tune on A2T dataset".

3 Experiment One: Can we teach a model twice?

3.1 Experimental procedure

For each task, we compare the performance and training efficiency of directly fine-tuning the pre-trained mT5-small model with that of fine-tuning model the have been tuned in other two dataset.

3.2 Experimental settings

We use mT5 [5] from huggingface as our model, which is a text-to-text model pre-trained on mC4 [3]. Because we only have one RTX-3090 with 24GB memory for training, we use mt5-small in our experiment, which has 300,176,768 parameters. The hyper parameters in each task are as follows:

- **A2T:** We set learning rate to 5e-4 and total number of epochs to 10. And the task prefix is set as "abstract to title: "
- **Translation:** We set learning rate to 5e-4 and total number of epochs to 10. And the task prefix is set as "translate English to Chinese: ".
- **Review classification:** We will set learning rate to 7e-4 and the total number of epochs to 100. The task prefix is set as "multi-label classification: "

Besides, we set batch size to 16 in all of the experiment. For other hyper parameters, we use the default settings of huggingface trainer [3].

3.3 Results

3.3.1 A2T

Tab. 1 shows the performance in testing set, and Fig. 2 shows the training loss. We can see that there is no significant difference in the final performance and training convergence speed between models fine-tuned once and models fine-tuned twice in this task.

3.3.2 Translation

Tab. 2 shows the performance in testing set, and Fig. 3 shows the training loss. Similarly, we find that there is no significant difference in the final performance and training convergence speed between models fine-tuned once and models fine-tuned twice in this task.

[3] https://huggingface.co/docs/transformers/main_classes/trainer

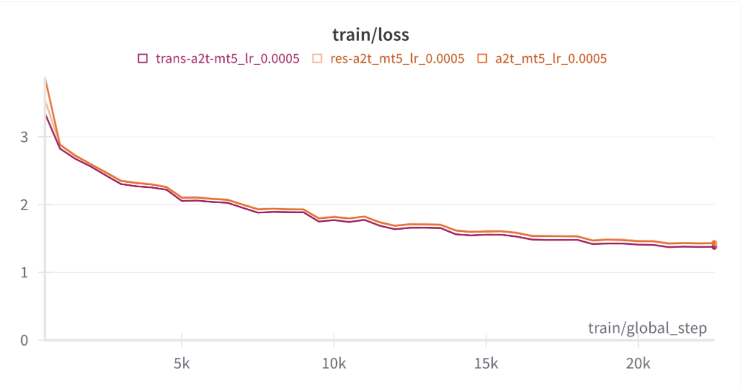


Figure 2: Training loss on A2T task.

Finetuned Datasets	sacrebleu
trans	**4.9516**
res -> trans	4.4340
A2T -> trans	4.7904

Table 2: Results on translation.

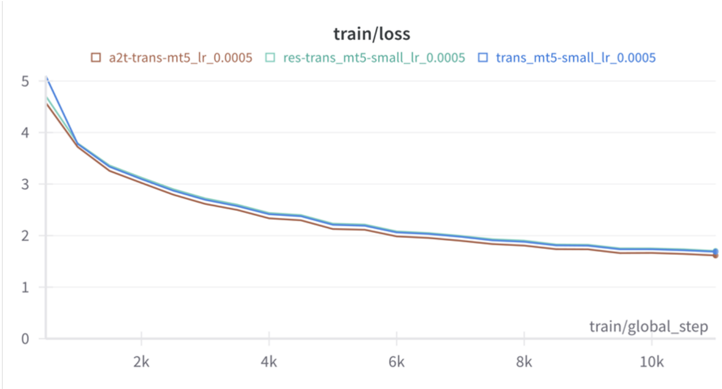


Figure 3: Training loss on translation task.

3.3.3 Restaurant review classification

Tab. 3 shows the performance in testing set, and Fig. 4 shows the training loss. We can see that models fine-tuned on the other two datasets converge much faster than the pre-trained mT5 model. Besides, although models that fine-tuned twice exhibit a lower training loss, their performance isn't significantly better than simply fine-tuning a pre-trained model directly. We suspect that this might be due to the relatively small size of the dataset, along with a vast number of label combinations in this multi-label classification task, which could lead to considerable differences between the test and training sets.

3.3.4 Analysis

In this experimental setting, teaching a model twice will not hurt its performance; it may even speed up its convergence. Besides, in my view, the similarity between restaurant review classification and translation is quite low, but it seems the model gets some insights from translation when doing classification. This "insight" might not be about any specific task itself, but about the form of output in downstream task. Additionally, it's worth noting that the restaurant review dataset is approximately 10 times smaller than the other two datasets.

Then we came up with this idea: Can we teach the pre-trained model a multitude of tasks simultaneously? This involves two questions: (i) Can the model retain its performance across these tasks as compared to being directly fine-tuned? (ii) Can the model perform better or converge more quickly on other tasks that it has not been trained on?

Finetuned Datasets	Accuracy	F1 score
res	0.3717	0.4742
A2T -> res	0.3717	**0.4751**
trans -> res	**0.3926**	0.4701

Table 3: Results on restaurant review classification.

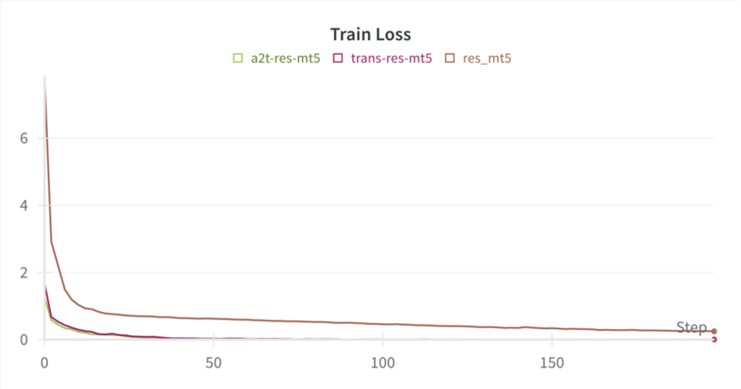


Figure 4: Training loss on restaurant review classification task.

4 Experiment Two: Can we teach a model various tasks simultaneously?

4.1 Fine-tune mT5-small on translation and A2T dataset simultaneously

We combine the A2T and Translation datasets at an approximate ratio of 2:1, then test on each dataset separately and fine-tune and test on restaurant review classification dataset. The fine-tuning parameter settings are the same as in Experiment One.

As shown in Tabs. 4a and 4b and Fig. 5a, the model is capable of learning both tasks simultaneously. However, when trained this way, its performance tends to be suboptimal, as evidenced by higher training losses and inferior results on testing set compared to those obtained from fine-tuning it directly on a specific task.

Besides, as shown in Fig. 5b and Tab. 4c, when fine-tuned on review classification dataset, the model that has been tuned on mixed dataset converged faster and performed better than the pre-trained model.

Finetuned Datasets	sacrebleu	rougeL	rouge1	rouge2
A2T	**11.5420**	**0.3701**	**0.4067**	**0.2304**
mixed dataset	9.3969	0.3290	0.3640	0.1893

(a) Results on A2T.

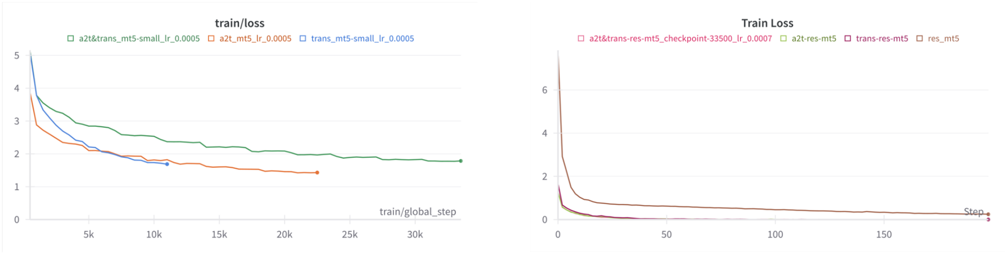
Finetuned Datasets	sacrebleu
trans	**4.9516**
mixed dataset	4.0750

(b) Results on translation.

Finetuned Datasets	Accuracy	F1 score
mT5, res	0.3717	0.4742
mT5, A2T -> res	0.3717	0.4751
mT5, trans -> res	0.3926	0.4701
mT5, mixed dataset -> res	**0.4031**	**0.5153**

(c) Results on restaurant review classification.

Table 4: Results of mixed training.



(a) Training loss on mix dataset.

(b) Training loss on restaurant review classification task.

Figure 5: Training loss of mixed training.

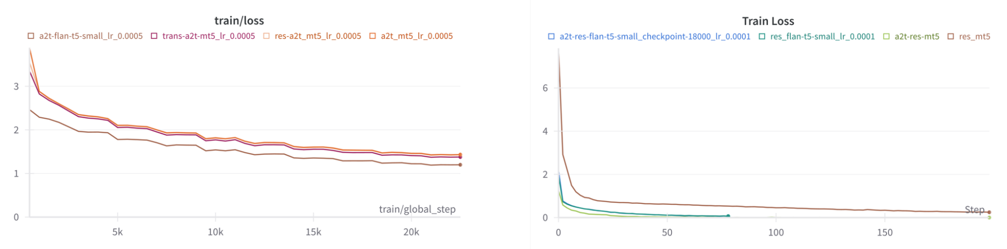
Finetuned Datasets	**sacrebleu**	**rougeL**	**rouge1**	**rouge2**
mT5, A2T	11.5420	0.3701	0.4067	0.2304
mT5, res -> A2T	11.6576	0.3708	0.4120	0.2306
mT5, trans -> A2T	11.1662	0.3675	0.4054	0.2263
Flan-T5, A2T	**11.8514**	**0.3748**	**0.4124**	**0.2310**

(a) Results on A2T.

Finetuned Datasets	**Accuracy**	**F1 score**
res	0.3717	0.4742
A2T -> res	0.3717	0.4751
trans -> res	0.3926	0.4701
Flan-T5, res	**0.5131**	**0.5427**
Flan-T5, A2T -> res	0.4240	0.5169

(b) Results on restaurant review classification.

Table 5: Comparison of performance of fine-tuning Flan-T5 and mT5 models.



(a) Training loss on A2T task.

(b) Training loss on restaurant review classification task.

Figure 6: Comparison of training loss of fine-tuning Flan-T5 and mT5 models.

4.2 Instruction finetuning

How about fine-tune the pre-trained model on a lot of downstream tasks simultaneously? Due to the limited time and computing resource, we directly leverage the Flan-T5-small model [1], which is based on T5 and has gone through instruction finetuning. Because the Flan-T5 tokenizer does not support Chinese [4], we only conduct experiment on A2T and review classification task.

As shown in Tab. 5 and Fig. 6, Flan-T5 consistently outperformed other models in both A2T and review classification tasks, with a particularly notable edge in the latter.

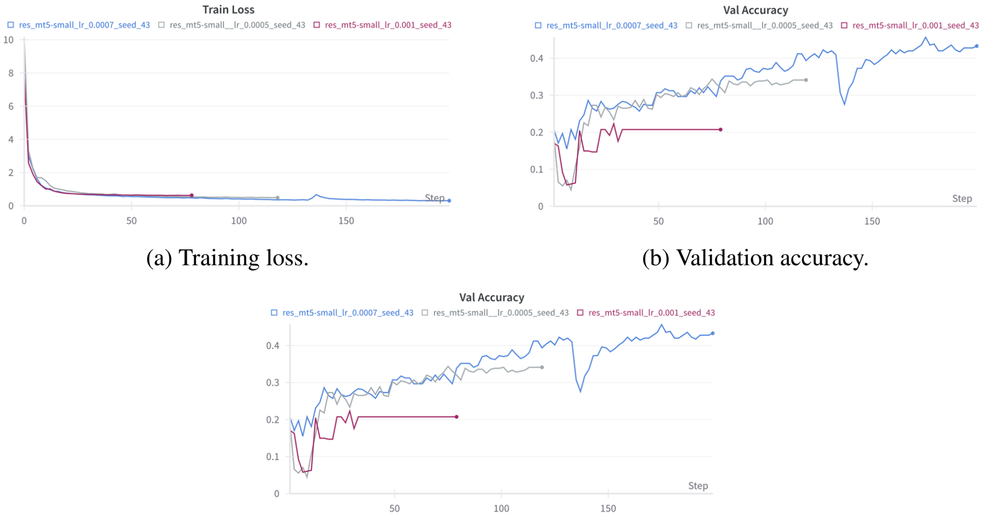
We also experimented with fine-tuning the Flan-T5-small model twice. However, as demonstrated in Fig. 6b, doing so leads to a decline in the model's performance.

5 Conclusion

Based on our experiment, we can deduce that (under this specific experimental settings):

- Models can be taught twice.

[4] https://huggingface.co/google/flan-t5-xxl/discussions/33



(a) Training loss.

(b) Validation accuracy.

(c) Validation F1 score.

Figure 7: Adjusting learning rate is quite tricky. For instance, in review classification task, when lr is set to 1e-3 (indicated by the pink line), we observe a decline in the loss; however, the accuracy plateaus, and the F1 score even drops to zero.

- Models can simultaneously learn multiple tasks, but its performance will be lower than directly fine-tuning on specific task. When encountering new tasks, they have the potential to exhibit improved performance and require less time for fine-tuning.
- For a pre-trained language model, the performance of instruction-tuning + fine-tuning on specific task > instruction-tuning + fine-tuning twice > fine-tuning twice >= fine-tuning once

Therefore, if my boss gives me several different tasks, I would directly fine-tune a LLM that has already undergone pre-training and instruction fine-tuning (e.g. Llama 2, Flan-T5) in different tasks separately.

6 Limitations

Our experiment has the following limitations:

- The datasets and models in the experiment are very limited.
- As shown in Fig. 7, adjusting learning rate is quite tricky, and I'm not sure my settings will be the optimal one.
- The test and validation sets for review classification are relatively small; moreover, due to the variance in the labels themselves, choosing the model with the best validation performance for testing could result in considerable error.

References

[1] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 5

[2] Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019. 1

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019. 2

[4] Hammam Riza, Michael Purwoadi, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Sethserey Sam, et al. Introduction of the asian language treebank. In *2016 Conference of The Oriental Chapter of International*

Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), pages 1–6. IEEE, 2016. 1

[5] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020. 2