Few-shot learning with parameter-efficient tuning

Yifei Song
École Polytechnique Fédérale de Lausanne
yifei.song@epfl.ch

Abstract

Our research explores the implementation of Parameter-Efficient Fine-Tuning (PEFT) and prompt tuning for few-shot learning adaptation of a pre-trained language model to specific downstream tasks. We initially evaluate the efficacy of PEFT on two popular, distinct model architectures, Encoder-Decoder and Decoder-Only, on publicly available datasets. We then transpose our findings to a real-world application on the Politosphere dataset. This study highlights the utility of PEFT and prompt tuning in few-shot learning and their adaptability to practical tasks, bridging the gap between academic research and real-world applications.

1 Introduction

With the emergence of large language models (LLMs), we have seen an impressive demonstration of their effectiveness across various natural language processing tasks. A multitude of pre-trained LLMs has been proposed, showcasing remarkable performance. However, despite these advancements, the need for extensive data to fine-tune these models for adaptation to specific downstream tasks remains a costly process. In response to this challenge, learning paradigms such as Zero-shot learning and Few-shot learning (Brown et al., 2020) have been proposed. These innovative methodologies offer a faster and more efficient approach to adapt LLMs to downstream tasks, forming the cornerstone of our research.

In our study, we investigate the specific impact of various factors, such as model architecture, model size, and training dataset size, on the performance of large models in downstream tasks within the context of few-shot learning. Our objective is to delve into the intricate interplay between these elements and how they shape the efficiency and effectiveness of few-shot learning in real-world applications.

2 Background

2.1 PEFT

Parameter-Efficient Fine-Tuning (PEFT) is a large-model training tool provided by Huggingface that offers a plethora of fine-tuning techniques such as LoRA and Prompt Tuning (Lester et al., 2021). This tool enables Pretrained Language Models (PLMs) to effectively adapt to a variety of downstream applications without necessitating fine-tuning of all model parameters, thereby significantly reducing training costs. In many NLP tasks, PEFT has already demonstrated performance on par with full fine-tuning. This efficiency is pivotal in our exploration of large language models and their adaptability in the context of few-shot learning.

2.2 Prompt tuning

Prompt tuning is another noteworthy technique in the realm of language model training. As a variant of fine-tuning, it uniquely positions itself by transforming an NLP task into a fill-in-the-blank format. Rather than making intricate adjustments to the entire model, prompt tuning focuses on optimizing a series of tokens, or prompts, which act as precursors to the target responses. This streamlined tuning process reduces the complexity and computational requirements of model adjustments, offering a cost-effective alternative to traditional fine-tuning methods.

3 Models

In our study, we delve into a comparative analysis of two distinct architectures of language models: Encoder-Decoder (Vaswani et al., 2017) and Decoder-Only (Liu et al., 2018). Each structure carries unique characteristics, providing a rich comparative basis for our experimentation.

3.1 Encoder-Decoder Models

This type of architecture is a two-step process model featuring an encoder that processes the input data and a decoder that generates the output. One of the hallmark examples of this structure in our study is the T5 series, including T5 (Raffel et al., 2020) and FlanT5 (Chung et al., 2022). The T5 models, built upon the Transformer architecture, have a reputation for their flexibility and performance across a broad spectrum of tasks. They first encode the input and then decode the output, benefiting from attention mechanisms that link the input and output at every stage of computation.

3.2 Decoder-Only Models

In contrast to the Encoder-Decoder architecture, Decoder-Only models consist of a single component responsible for processing the input and generating the output. A representative example from our experiment is the Bloomz series (Muennighoff et al., 2023). These models streamline the encoding-decoding process into a single step, reducing the computational complexity. Despite their simplicity, they maintain an impressive capability for various tasks, demonstrating the effectiveness of focused model architectures.

These two architectures serve as the primary foundation for our experimentation, allowing us to probe the impact of model structure on the performance of PEFT in few-shot learning scenarios. Their diverse characteristics offer us a comprehensive view of model behaviour under different parameter tuning techniques.

4 Datasets

Our study employs two distinct datasets: GLUE SST2 and Politosphere.

4.1 GLUE SST2

The General Language Understanding Evaluation (GLUE) (Wang et al., 2019) benchmark comprises several datasets for evaluating natural language understanding tasks. Among these, the Stanford Sentiment Treebank version 2 (SST2) is a popular dataset used for sentiment analysis. The SST2 dataset includes sentences from movie reviews along with their binary sentiment labels, categorized as positive or negative. This allows the models to learn from nuanced language cues to discern overall sentiment.

4.2 Politosphere

The Politosphere dataset we utilized originates from Reddit Forums, offering a rich collection of real-world discussions. More specifically, we selected subreddit forums with politically charged topics that are likely to elicit diverse opinions, with all threads being centered around gun control. To ensure high data quality and clear-cut viewpoints, we handpicked data from three subreddit forums: "progun," "liberalgunowners," and "guncontrol."

Scores. We define the score of each comment as the difference between the number of likes and dislikes received from users. For dataset construction, we selected a portion of the highest-scoring data in each subforum, such as the top 0.9 quantile.

Communities. By calculating the average scores of each user's posts across different subreddits, we obtain vector representations for users. Leveraging unsupervised learning techniques (Louvain algorithm), we cluster these users to define Communities.

Soft Community Score. Given that a user or comment might simultaneously belong to several communities, we utilize a probability distribution vector for community representation. This distribution sums up the likelihood of each user's affiliation to each community, presented in the form of:

$$[c_1, c_2, \cdots, c_8]$$

We assume that there are 8 communities.

This approach helps mitigate potential information loss caused by relying solely on the maximum value.

Hard Community Score. To facilitate the creation of a supervised dataset, we designate the majority community in the Soft Community Score as the comment's label, also referred to as the Hard Community Score.

The most representative users from each community, those with the highest predicted soft community scores, are chosen to form the core of the training set. A small portion of these users are further set aside for evaluation purposes. Additionally, we create a 'val_polarized' set containing users who exhibit strong polarization beyond the primary training set. The remaining users are divided into a 'val' development set and a 'test' set, allowing evaluation across a spectrum of users, both polarized and unpolarized.

Each comment in the dataset is associated with its predicted hard community, and separate files are

saved for each split ('train', 'val_polarized', 'val', 'test') for flexible usage.

The use of both GLUE SST2 and Politosphere provides a robust platform for evaluating our models. While SST2 offers a general and widely recognized benchmark, Politosphere provides a customized, real-world scenario for assessing the effectiveness of our few-shot learning approach.

5 Evaluation

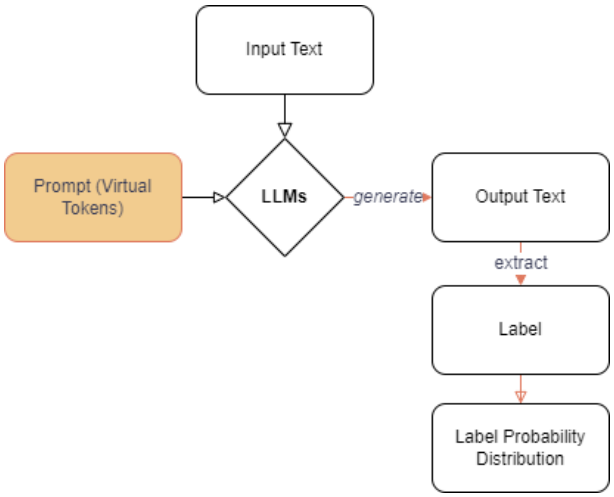

Figure 1: Flowchart depicting the prompt tuning process for a large-scale language model.

In our research, we primarily employ the method of pre-training the language model through prompt tuning. Subsequently, we leverage the text-generating property of the model to produce prediction labels, which we compare against the ground truth (Figure 1). Our evaluation metrics include loss, accuracy, precision, recall, F1 score and similarity.

The GLUE SST2 dataset, recognized for its high quality and cleanliness, primarily necessitates a keen focus on the accuracy of the model output. However, in evaluating the performance on the Politosphere dataset, we broaden our evaluative scope. We take into account all of these metrics for both users and comments, acknowledging the potential influence of factors such as data distribution and quality.

Similarity. Cosine similarity between the per-user soft community scores, predicted by graph clustering and the per-comment soft community scores predicted by the supervised classifier, and averaged on all user's comments.

Outliers. In situations of small-sample training, our models may produce outliers - values outside of known categories. We handle these by assigning them to undefined categories and converting model outputs to probability distributions of predicted categories, rather than direct outputs. This facilitates better metric computation and helps monitor the influence of undefined categories on predictions.

6 Experiments and Results

6.1 T5 and Flan-T5

Our experiment involved testing the effects of prompt tuning with PEFT on the GLUE SST2 dataset with T5 and its fine-tuned variant, Flan-T5. We initialized the prompt text as "Classify if the sentence is positive or negative:", and set the length of virtual tokens to 12.

Prompt tuning was initially carried out on the full training set with T5 series models of varying sizes, and we observed from Figure 2 that:

- Flan-T5 outperformed the original T5 model in few-shot learning. Even the smaller version, Flan-T5-Small, delivered satisfactory results from the onset. This superior performance is likely due to Flan-T5 building upon the pre-trained T5 model, which results in improved zero-shot learning capabilities.

- The performance of the models correlated positively with their size. It was challenging for smaller models to surpass their larger counterparts within the same series, even with the application of prompt tuning.

- The T5-Base model, once subject to prompt tuning, was able to achieve results comparable to the Flan-T5-Base model in the sentiment classification task.

- Models with insufficient pre-training, such as T5, tended to underperform in the early stages of prompt tuning. This was more evident in larger models, like T5-Large, compared to T5-Base. The probable reason is the limited number of parameters during prompt tuning training, which constrains the model's ability to swiftly adapt to new downstream tasks with a small amount of training data.

Based on our previous experiments, it is evident that although Flan-T5 exhibits superior overall performance, its trainability is comparatively limited. Therefore, we proceeded to investigate the impact

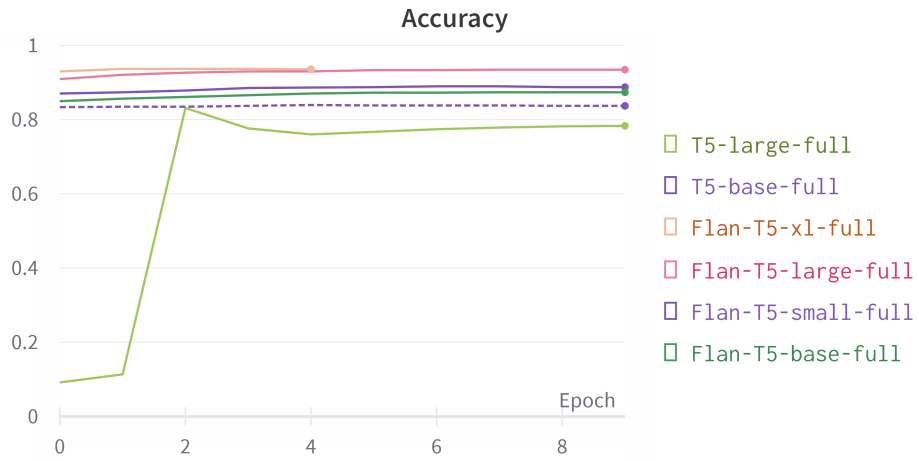

Figure 2: Predictive accuracy of Flan-T5-Small (80MB), Flan-T5-Base (250MB), Flan-T5-Large (780M), Flan-T5-XL (3B) and T5-Base (60MB), T5-Large (770MB) for prompt tuning on the full training set of SST2. Flan-T5-XL achieves the highest accuracy with a value of 0.94.

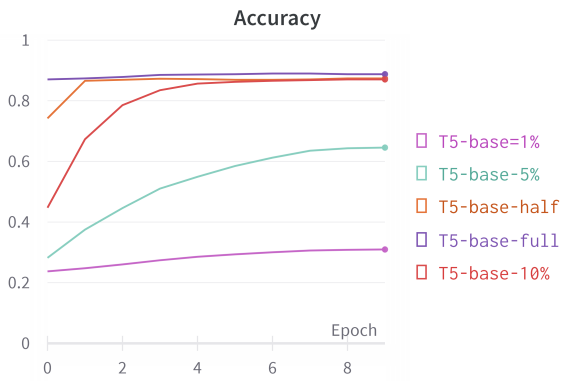

Figure 3: Predictive Accuracy of Flan-T5-Base (250MB) on Different-Sized Training Sets of SST2. T5-base-full achieves the highest accuracy with a value of 0.89.

of different-sized training sets on the Prompt Tuning results of the T5 model. As illustrated in Figure 3, the following observations can be made:

- Larger models demonstrate better performance during the initial stages of training as well as overall.
- With just 10% of the training set, prompt tuning achieves comparable results to those obtained with the full training set.
- When the training set is excessively small, the training efficacy is significantly compromised.

6.2 Bloomz

We proceeded to evaluate the performance of Bloomz on GLUE SST2 through prompt tuning. To ensure comparability with the previous T5 series models, we aimed for similar trainable parameters during prompt tuning. We maintained the same initial prompt text and virtual token lengths as in the previous experiments. Notably, we observed from Table 1 that T5-Large, Flan-T5-Large, and Bloomz-560m had identical trainable parameters after freezing the model weights and incorporating virtual tokens.

Table 1: Trainable Parameters of Different Models during Prompt Tuning with Virtual Token Length of 12

Model Name	Trainable Params	All Params
T5 base	9K	222M
Flan T5 base	9K	248M
T5 large	12K	783M
Bloomz-560m	12K	559M
Bloomz-1b1	18K	1.1B

Therefore, we incorporated T5-Large and Flan-T5-Large into the performance evaluation of the Bloomz model. Prompt tuning was conducted on the full training set, and the following observations were made from Figure 4:

- Bloomz models exhibited superior overall performance compared to T5, albeit slightly inferior to Flan-T5.

- Large pre-trained models, such as Bloomz-1b7, a similar phenomenon was observed as with T5-Large, where a low number of virtual tokens hindered the model's rapid adaptation to new downstream tasks.

- Different sizes of Bloomz models demonstrated relatively comparable performance on the SST2 sentiment classification task after prompt tuning.

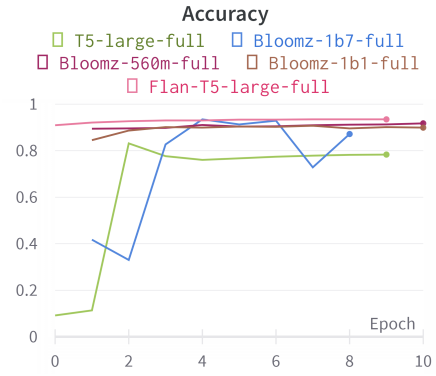

Figure 4: Predictive Accuracy of Bloomz-560m, Bloomz-1b1, Bloomz-1b7, Flan-T5-Large (780M) and T5-Large (770MB) on the full Training Sets of SST2. Flan-T5-Large achieves the highest accuracy with a value of 0.92.

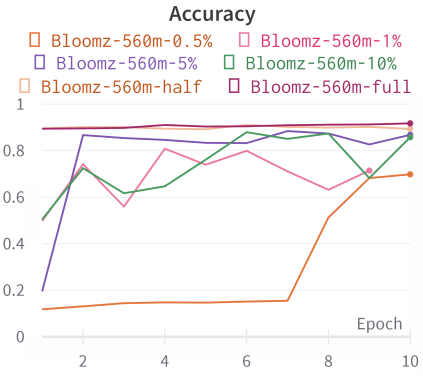

Figure 5: Predictive Accuracy of Bloomz-560m on Different-Sized Training Sets of SST2. Bloomz-560m-full achieves the highest accuracy with a value of 0.91.

To further probe the few-shot learning capabilities of Bloomz, we tested Bloomz-560m using various sizes of training sets. In parallel, we also subjected T5-Large and Flan-T5-Large models, with an equal number of training parameters, to the same testing procedure. From Figures 5 and 6, we derived the following observations:

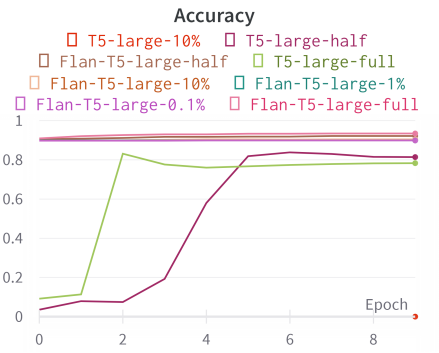

Figure 6: Predictive Accuracy of T5-Large and Flan-T5-Large on Different-Sized Training Sets of SST2. Flan-T5-Large-full achieves the highest accuracy with a value of 0.93.

- During few-shot learning with prompt tuning, Bloomz-560m's performance displayed significant fluctuations, yet the overall accuracy trend was upward.

- Bloomz-560m demonstrated better performance as the training set size increased. Nevertheless, even with just 0.5% of the training set, the model was still capable of achieving over 60% accuracy through few-shot learning.

- Due to its superior zero-shot learning capabilities, Flan-T5-Large's performance seemed largely uninfluenced by the size of the training set, nor did it exhibit notable improvement. Given that the training task was relatively straightforward, the model's potential for few-shot learning in more complex downstream tasks remains an open question.

- Compared to T5-Large, Bloomz-560m displayed markedly superior few-shot learning capabilities under the condition of equivalent training parameters. When the training set was reduced to 10%, T5-Large's performance even dropped to 0% accuracy, while Bloomz-560m maintained decent performance even with smaller training sets.

6.3 Politosphere

From our preceding experiments, we observed that Bloomz demonstrated superior performance in few-shot learning, and it offered larger room for training improvement. Therefore, we conducted tests on Bloomz-560m using the real-world dataset, Politosphere. Unlike the sentiment classification in

Table 2: Metrics of Bloomz-560m on Politosphere Dataset.

Comment Metrics			
Accuracy	F1	Precision	Recall
0.495	0.260	[0.345, 0.571, 0.502]	[0.070, 0.002, 0.963]

User Metrics				
Accuracy	F1	Precision	Recall	Similarity
0.33	0.205	[0.344, 0.500, 0.333]	[0.071, 0.004, 0.927]	0.339

SST2, the current labels for prediction are more diversified. As can be observed in Table 2, our experimental results to date are not entirely satisfactory, with the model tending to predict text into a fixed category. The possible reasons for this outcome and the improvement measures will be discussed in the following sections.

7 Future Work

In our current work, our performance on the Politosphere dataset leaves room for improvement. Several underlying reasons could potentially contribute to this, alongside corresponding measures for refinement:

1. Bloomz-560m may not possess sufficient parameters to learn more complex relationships from the training set. Therefore, we plan to experiment with larger models, preferably those with at least 3 billion parameters.

2. Our current results exhibit clear imbalances, likely due to the unbalanced distribution within the dataset. The current proportion among different classes is 1:2:3. We could ameliorate this by selecting more representative data to achieve a balanced distribution among various classes as much as possible.

3. The quality of the current dataset may not be high enough, with some data inputs demonstrating weak correlations with their corresponding labels, thus making it challenging for machines to learn the classification. Given the model's promising few-shot learning capability, we propose selecting a small number (in the hundreds) of high-quality data as the training set for the model to learn.

4. We intend to explore the use of more advanced Decoder-Only models, such as LLaMa-7B (Touvron et al., 2023) and LLaMa-Alpaca (Wang et al., 2023), in our future research.

8 Conclusion

Our research on parameter-efficient fine-tuning and prompt tuning for few-shot learning has yielded valuable insights into the adaptability and performance of large language models. Through experiments on T5, Flan-T5, and Bloomz models, we have observed the benefits of prompt tuning and the impact of model architecture, size, and training set size on few-shot learning. Our findings demonstrate the superiority of Flan-T5 models over T5 models, especially in the early stages of training. We have also confirmed the potential of Bloomz models, such as Bloomz-560m, in achieving satisfactory performance with smaller training sets. However, further improvements are needed to address challenges in real-world datasets, including model scalability, dataset balance, and data quality. Future research will focus on exploring larger models, handling dataset imbalances, selecting high-quality data, and investigating advanced Decoder-Only models. Our study contributes to the advancement of few-shot learning techniques and the practical application of language models in various tasks.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions.