Evaluation of Galactica LLM on Biological Tasks with Parameter-Efficient Fine-Tuning Methods

Ryan Engel and Gilchan Park

Computational Science Initiative
Brookhaven National Laboratory

Abstract

In the sphere of natural language processing, the paradigm of pre-training Large Language Models (LLMs) on broad domain data, followed by task-specific adaptation, has shown immense potential. However, due to the inherent complexity of gene/protein relationships in molecular biology, applying such technology to this specific domain presents unique challenges. In this project we aim to overcome such challenges by implementing state of the art prompt engineering strategies, which will further advance our ability to use LLMs for understanding biomolecular data. Our efforts have yielded improvements that outshine contemporary prompting strategies, thus validating the efficacy of our approach. The use of advanced techniques such as parameter-efficient prompt tuning and low rank adaptation, has further optimized our use of the Galactica large language model, pushing the boundaries of what's achievable in this domain. This work marks a significant stride in integrating artificial intelligence with molecular biology, paving the way for rapid advancements in life sciences and healthcare.

1 Introduction

Scientific research often necessitates deciphering vast amounts of complex data. Although LLMs, such as Galactica from Meta AI, have the potential to assist in this task, it is computationally expensive to fine-tune such models to domain-specific tasks, and the complexity of biomedical data exacerbates this challenge. Problems such as medical question-answering or scientific named entity recognition require specialized knowledge that may not be fully captured by general-purpose LLMs. In this project we aim to address these issues and propose viable solutions to these tasks by implementing strategies that have shown promising results in the field. By focusing on novel prompt engineering strategies for LLMs, we streamline the process of model optimization. This work provides the way for improved understanding and extraction of information from complex biological datasets using LLM, offering a more resource-efficient approach to data analysis in the realm of molecular biology.

2 Hypothesis

We hypothesized that by implementing cutting-edge prompt engineering strategies such as parameter-efficient prompt tuning and low rank adaptation, we could significantly enhance the capabilities of LLMs like Galactica in interpreting and extracting valuable insights from scientific datasets. In particular, we expected our approach to outperform traditional methodologies, resulting in more efficient extraction of complex biomolecular data from sources such as the PubMedQA dataset and the STRING database. Our findings have provided substantial evidence in support of this hypothesis, indicating that strategic implementation of modern NLP techniques has the potential to transform the landscape of data analysis in molecular biology.

3 Methodology

3.1 Overview

The project's experimental framework was strategically designed to employ a multi-faceted approach for evaluating and refining the application of LLMs. The foundation of this process was anchored on the application of cutting-edge prompt engineering strategies, specifically parameter-efficient prompt tuning and low-rank adaptation (LoRA), to Galactica from Meta AI.

The Galactica models are trained on a large-scale scientific corpus. They are designed to perform scientific tasks, including citation prediction, scientific QA, mathematical reasoning, summarization, document generation, molecular property prediction, entity extraction, and more. The experiment harnessed the computational efficiency of the Galactica-mini version, which encapsulates a more compact representation of 125 million parameters. This variant of Galactica was employed to facilitate a more resource-conservative analysis, while still utilizing the capabilities of the model.

3.2 Data Preprocessing and Model Training

Initial stages involved the preprocessing of the selected datasets, PubMedQA and the STRING database, facilitated through a Python script. The training process involved two primary configurations: prompt tuning and LoRA, both utilizing a neural network as the core architecture for the model optimization. We conducted the performance comparison of both configurations, and their effectiveness was also evaluated against a control group, comprising the identical model and datasets without prompt engineering schemes.

3.3 PubMedQA Dataset

PubMedQA is a biomedical question answering (QA) dataset collected from PubMed abstracts. Its task is to answer biomedical questions with yes, no,

or maybe, using corresponding context. PubMedQA has 1k expert-annotated, 61.2k unlabeled and 211.3k artificially generated QA instances. For this project we used only the 1k expert-annotated QA instances to reduce cost of computing power.

Each question contains some context that can be used to give more information to the model, and the correct answer to that question. We split up this data into 3 categories, the first 450 instances were used as the training data, the next 50 were used as validation data, and the last 500 instances were used for testing and model evaluation. Here is an example of one instance of the dataset:

Question: "Is there a connection between sublingual varices and hypertension?"

Context: "Sublingual varices have earlier been related to ageing, smoking and cardiovascular disease. The aim of this study was to investigate whether sublingual varices are related to presence of hypertension. Etc."

Answer: "Yes"

3.4 Complications with the PubmedQA Dataset

During our evaluation of the PubMedQA dataset, we encountered intricacies in its data structure. Each data entry in this set encompasses a question and its corresponding answer. Additionally, supplementary information is provided, namely: "context", "long answer", and flags termed "reasoning free" and "reasoning required". The precise utilization of these parameters was initially ambiguous. However, after an in-depth analysis, the following protocol was established.

For any given question, if the "reasoning free" parameter is true, the "long answer" is coupled with the question, facilitating the model in directly extracting the answer from the text. Conversely, if the "reasoning required" flag is true, the "context" is combined with the question. This approach compensates for the model's inability to derive answers as straightforwardly as it does from the "long answer", thus necessitating augmented contextual information. Notably, the "context" field is typically more comprehensive than the "long answer", providing a richer backdrop for the model.

In instances where a question is flagged with both "reasoning free" and "reasoning required", the question is merged with both the context and the long answer. Conversely, if both flags register as "no" or "maybe", the corresponding long answer and context are excluded from the input data. Implementing this methodology notably improved the model's performance.

3.5 STRING Database

STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) is a biological database and web resource of known and predicted protein–protein interactions. First, we preprocessed the data into the form of a question and an answer. Then we randomize the data in the set, and pick the first 1,000 question-answer pairs. Next, we split this data up into 70% training, 15% validation, and 15% testing. Here is one example of the data:

Question: Which proteins are related to MRPL52?

Answer: ATP5O, C3orf2, C9orf72, ENSG26266, GDE1, HSPBP1, ICT1, MRPL18, MRPL24, MRPL32, MRPL39, MRPL46, MRPL49, MRPL54, MRPS18C, MRPS23, MRPS24, MRPS36, NDUFS1, NOP1, PCP2, RPL14, RPL26L1, RPL36, RPS18, RPS27L, RPSA, SLC25A1, SULT1A4, ZBED1

4 Evaluation and Analysis

We evaluated the model performances by F1 score and accuracy for the PubMedQA dataset, which focuses on the task of biomedical question answering (yes/no/maybe). For the STRING database, token precision was employed as the main evaluation metric, which is based on the number of matches between model's generated proteins and the same number of true proteins.

4.1 Text Classification for PubMedQA Data

Given our utilization of a causal language model and our approach to question answering as a text generation task, it becomes imperative to meticulously evaluate the resultant outputs and to accurately compute the F1 and accuracy metrics, particularly for the PubMedQA data. To this end, we initiate the process by channeling the input sequence through the model. Subsequently, the model's output is procured using the k-sampling for few-shot strategy. It's noteworthy to mention that our optimal results, delineated in the subsequent sections, were obtained with $k = 1$. However, various k-values were also probed as potential hyperparameters.

For the classification of the model output, we employed the VADER (Valence Aware Dictionary and sEntiment Reasoner) text classifier available from HuggingFace. VADER is a rule-based sentiment analysis tool optimized for sentiments prevalent in social media contexts. Despite its design, we ascertained its efficacy in analyzing and classifying our model's results. This methodology categorizes a text as positive, negative, or neutral based on a distinct score computed by the tool.

Through rigorous experimentation, we deduced that a compound polarity score of $\geq .45$ warranted a "yes" classification, while scores $\leq .45$ were designated

as "no". Scores that reside within this range, reflecting ambiguity or lack of definitive classification, were labeled as "maybe".

4.2 Analytical Observations for STRING Data

The endeavor to utilize text generation for answers pertaining to the STRING dataset presents inherent complexities. Upon reflection, this approach might not be the optimal method for harnessing the potential of this dataset. The primary objective was to train the model in text generation, and by assessing the token-level precision between the generated and actual answers, infer the model's proficiency in protein prediction.

However, this method encountered challenges. Notably, the model exhibited tendencies to produce spurious outputs, including repetitive sequences of specific numerals and characters. Nonetheless, this study highlighted that parameter-efficient fine-tuning strategies outperformed conventional prompting techniques.

5 Model Architecture

5.1 Prompt Tuning

Prompt tuning incorporates the concept of soft prompts to condition pre-trained language models, to execute specific tasks. Differing from traditional model architectures that use discrete text prompts, soft prompts are trained via back-propagation, allowing adaptability based on labeled data samples.

As a model scales into billions of parameters, prompt tuning exhibits superior performance, rivaling the results of comprehensive model tuning wherein all weights are adjusted. This technique is particularly advantageous for large models, demanding significant resources for distribution and operation. As such, a single pre-trained model can be repurposed for various tasks using prompt tuning.

Traditional classification, modeled as $Pr(y|X)$, where X is a series of tokens and y is a single class label, is transformed into conditional generation. Now, Y is a sequence of tokens that represent a class label, but with prompt tuning we model this classification as $Pr_\theta(Y|X)$, parameterized by the transformer weights θ.

Prompting is essentially the addition of extra tokens, P, to the input X to condition the model during the generation of Y. In models like Galactica, the representations of the prompt tokens $P = \{p_1, p_2, \ldots, p_n\}$ are part of the embedding table, with parameters frozen at θ. Finding optimal prompts traditionally requires either manual search or non-differentiable search methods. However, prompt tuning alleviates this by introducing dedicated parameters θ_P

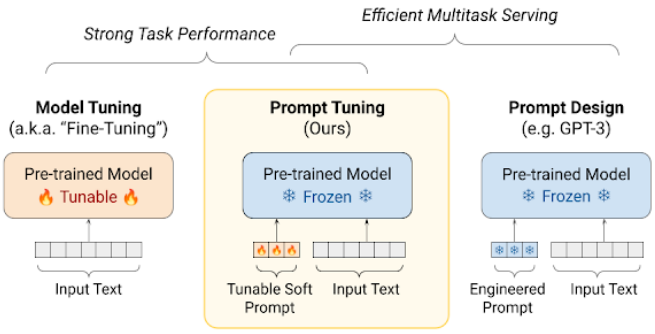

Figure 1: Prompt Tuning Diagram. Source (2)

for the prompt which can be updated. Thus, instead of selecting from a fixed vocabulary, prompt tuning adjusts the embeddings of these tokens. The new conditional generation is modeled as $Pr_{\theta,\theta_P}(Y|[P;X])$ where only θ_P gets updated during training.

Given a sequence of n tokens, $\{x_1, x_2, \ldots, x_n\}$, this strategy begins by embedding these tokens to form an embedding matrix $X_e \in R^{n\times e}$, where e is the dimension of the embedding space. The soft-prompts are defined as a parameter $P_e \in R^{p\times e}$, where p is the length of the prompt. This prompt is then concatenated with the embedded input to produce a matrix $[P_e; X_e] \in R^{(p+n)\times e}$. This matrix is then used in the typical encoder-decoder structure. Only the prompt parameters P_e are updated during training. This can be seen in the diagram above (figure 1).

Given a token series, we start by embedding the tokens, resulting in matrices to represent the token embeddings and soft-prompts. Following this, the prompt concatenates with the input, forming a combined matrix that integrates with the encoder-decoder mechanism. Importantly, only the prompt parameters undergo modifications during training. This is represented by the following equation:

$$\max_{\theta_P} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log\left(Pr_{\theta;\theta_P}\left(y_t|[P;x], y_{<t}\right)\right) \tag{1}$$

In this formula, θ_P denotes the tunable parameters of the prompt, and $[P;x]$ represents the concatenation of the prompt with the input sequence. $Pr_{\theta;\theta_P}$ gives the likelihood of the target series y, given the prompt and input sequence x, influenced by both the primary model parameters θ and the prompt parameters θ_P. Summation indices encapsulate all training instances (x, y) in the dataset Z

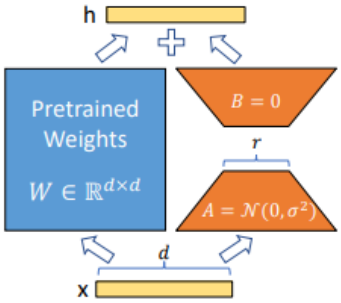

Figure 2: LoRA Reparameterization. Source (1)

and all tokens y_t in the target series y. The notation $y_{<t}$ represents the target sequence up to the $t-1$th token.

5.2 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) presents an innovative strategy for adapting large-scale pre-trained models to specific tasks or domains without necessitating full fine-tuning. By harnessing the inherent low-rank structure during adaptation, LoRA substantially diminishes the quantity of trainable parameters and computational overhead, rendering the fine-tuning of large language models like Galactica more feasible and efficient.

Traditional deep learning models typically comprise dense layers that execute matrix multiplication with full-rank weight matrices. Nevertheless, these matrices often display a low "intrinsic rank" during the adaptation phase. For a pre-trained weight matrix $W_0 \in R^{d \times k}$, its modification is delineated with a low-rank decomposition:

$$W_0 + \Delta W = W_0 + BA$$

where $B \in R^{d \times r}$, $A \in R^{r \times k}$, and $r \ll \min(d, k)$. Here, W_0 remains unchanged during training, while both A and B are trainable. This idea can be illustrated through the graphical representation above in figure 2. Given an input x, the modified forward pass is described as:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

LoRA advances the concept of fine-tuning by abstaining from enforcing the cumulative gradient update to weight matrices to possess full rank during the adaptation. As the LoRA rank r aligns with the rank of the pre-trained weight matrices, it emulates the expressiveness of thorough fine-tuning.

Upon deployment, the weight matrix $W = W_0 + BA$ is computed and stored,

ensuring standard inference without any additional latency. Switching to a different task mandates a simple subtraction and addition operation, guaranteeing no extra inference latency compared to conventionally fine-tuned models.

Incorporating the objective function, the model adaptation is mathematically represented as:

$$\max_{\Theta} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)}\left(y_t|x, y_{<t}\right)\right) \tag{2}$$

A prominent limitation of exhaustive fine-tuning is the necessity to assimilate a unique set of parameters, denoted as $\Delta\Phi$, for every downstream task. The dimensionality of $\Delta\Phi$ equates to that of Φ_0. Thus, if the pre-trained model is voluminous, deploying several iterations of independently fine-tuned models can metamorphose into a challenging or potentially impracticable endeavor.

Within our research, we adopt a more parameter-conservative method, whereby the task-specific parameter increment $\Delta\Phi = \Delta\Phi(\Theta)$ is further encoded via a notably reduced set of parameters, symbolized as Θ, with $|\Theta| \ll |\Phi_0|$. Consequently, the task of deducing $\Delta\Phi$ is transmuted into optimization over Θ. Here, the objective remains the identification of parameters Θ that amplify the likelihood of the target sequences, considering the input sequences, within the model parameters $\Phi_0 + \Delta\Phi(\Theta)$.

LoRA unfolds as a potent and efficient methodology for tailoring large pre-trained models to distinct tasks. By capitalizing on the low-rank structure embedded within the adaptation mechanism, LoRA offers an apt resolution to the dilemmas of computational and memory overheads accompanying the fine-tuning of colossal models.

6 Results

The experiments conducted within this research present compelling evidence that both parameter-efficient prompt tuning and low-rank adaptation of LLMs surpass the effectiveness of the conventional pre-trained Galactica model. The results indicate a distinctive operational advantage of each strategy within different contexts.

For instance, parameter-efficient prompt tuning displayed superior performance in the accuracy of answer prediction for questions derived from the PubMedQA dataset. Conversely, the LoRA configuration demonstrated enhanced effectiveness for the fine-tuning process in the generative task in the STRING dataset. Furthermore, the combination of both configurations has shown to be effective as well. Detailed results from each experimental trial are represented in the accompanying graphical illustrations. This visual data representation provides

a clear comparative view of the performance metrics of the various strategies implemented in the study.

Configuration Strategy	F1 Score	Accuracy
Original	54.75%	59.20%
LoRA	71.14%	69.30%
Prompt Tuning	**75.52%**	**77.40%**
LoRA + Prompt Tuning	70.98%	69.00%

Table 1: PubMedQA dataset results: All trials in this experiment were conducted using a batch size of 4. The learning rate for the LoRA configuration was 0.00001 and the learning rate of the other configuration strategies was 0.0001. For LoRA, the highest metrics were achieved with 4 epochs of training. For prompt tuning the highest metrics were achieved with 9 epochs of training. And for the combination of both strategies, the highest metrics were achieved with 8 epochs of training.

Configuration Strategy	Precision
Original	4.06%
LoRA	10.18%
Prompt Tuning	8.72%
LoRA + Prompt Tuning	**12.09%**

Table 2: String dataset results: Each of these experiments were conducted with a learning rate of 0.001, a batch size of 4, and 15 epochs of training, with the exception of the combined strategy, which was achieved with only 3 epochs of training.

7 Discussion and Conclusion

This research encapsulates a comprehensive exploration into the realm of Natural Language Processing, focusing specifically on the optimization of Large Language Models (LLMs) for the intricate domain of molecular biology. By employing novel prompt engineering strategies, we have been able to enhance the capabilities of the Galactica model from Meta AI, achieving marked improvements in data analysis for the selected biomedical datasets.

However, it is imperative to highlight the constraints posed by computational costs and resources in this endeavor. The demanding nature of our methodologies, necessitates substantial GPU usage and prolonged training periods. These factors currently limit the scalability and speed of our approach, potentially slowing down the rate at which valuable insights can be gleaned from vast scientific datasets.

Moreover, it is worth noting that the Galactica-mini variant used in our study is a more compact representation of the full model, chosen for its resource-efficiency. This suggests that our results, while promising, may only represent a subset of the potential performance achievable with larger, more computationally intensive models. This can be attributed to the fact that the parameter-efficient strategies implemented show better improvements when used on models with more parameters.

Despite these limitations, our results signify a crucial step forward in the integration of artificial intelligence with molecular biology. The success of our approach indicates that with further optimization and more efficient computational strategies, LLMs could become instrumental tools in the domain of life sciences.

Looking ahead, future work in this area should aim to mitigate the computational demands of these models while maximizing their predictive capabilities. The exploration of new methodologies and optimization techniques could lead to significant advancements, fostering more efficient and accurate extraction of information from complex biological datasets.

Furthermore, expanding the implementation of prompt tuning and LoRA strategies across different domains will allow us to assess their versatility and adaptability. The insights gathered from these expanded studies will further refine our understanding of these strategies, opening up new avenues for research and innovation in natural language processing.

In conclusion, the compelling results from our study highlight the potential of LLMs in molecular biology, offering promising prospects for future research in the biomedical domain. With sustained efforts in this direction, we anticipate rapid advancements in life sciences, medicine, and healthcare, powered by the transformative capabilities of artificial intelligence.

8 Related Work

1.) Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint. https://arxiv.org/pdf/2106.09685.pdf

2.) Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for computational linguistics. https://aclanthology.org/2021.emnlp-main.243.pdf

3.) Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the

Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). Association for computational linguistics. https://aclanthology.org/2021.acl-long.353.pdf

4.) Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv preprint. https://arxiv.org/pdf/2107.13586.pdf

5.) Guo, Z., Wang, Y., Wang, P., & Yu, S. (2023). Improving Small Language Models on PubMedQA via Generative Data Augmentation. MIT Department of Electrical Engineering and Computer Science.
https://arxiv.org/pdf/2305.07804.pdf

6.) Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N. T., Legeay, M., Fang, T., Bork, P., Jensen, L. J., & von Mering, C. (2021). The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Research, 49(Database issue), D605–D612.
https://doi.org/10.1093/nar/gkaa1074

9 Acknowledgements

I wish to express gratitude to my mentor, Dr. Gilchan Park, whose extensive expertise, understanding, and patience have added immeasurably to my experience throughout the Science Undergraduate Laboratory Internships (SULI) program. His guidance has been crucial to the success of this project.

I would also like to extend my thanks to Brookhaven National Laboratory and its Office of Educational Programs for offering an effective research environment. Additionally, I am grateful for the opportunity to work in the Computational Science Initiative at Brookhaven National Laboratory. Their vast computational resources have provided me with the means to take on this endeavor.