Efficient Domain-Specific Language Models: Fine-Tuning Gemma-2b-it for Data Science*

Name (SUNet ID): Aline Menezes (alinemsm)
Mentor: Rohan Taori
Collaborators: GPT-4, Claude-Opus

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in various natural language processing tasks. However, these state-of-the-art models often require significant computational resources and are prone to hallucination and inconsistency in their responses, particularly when dealing with niche domains that require specialised knowledge. To address these limitations, we propose a framework that converts specialised documents into customised, domain-specific language models by fine-tuning small, open-source models using curated domain knowledge.

In this paper, we focus on the field of data science, leveraging the abundance of high-quality, specialised knowledge available in widely accepted and reliable textbooks. We chose Gemma-2b-it as the base model due to its computational efficiency, strong performance with Low Rank Adaptation (LoRA) fine-tuning, and permissive open-source license. To create the dataset for fine-tuning, we leverage the capabilities of Claude-3 Opus to generate a synthetic dataset of questions and answers based on the selected textbooks.

We evaluate the performance of our fine-tuned model on a held-out set of Q&As and compute automated metrics to assess its quality and effectiveness. By comparing the model's performance against baselines and a state-of-the-art oracle, we aim to demonstrate the potential of our framework to create cost-effective and efficient domain-specific language models with limited resources. Our results show that the fine-tuned models significantly outperform the baseline, indicating promise in our approach. However, our error analysis reveals some limitations and areas for improvement, such as the need for more diverse training data, exploration of alternative evaluation metrics, and the potential benefits of incorporating expert feedback. In conclusion, our framework shows potential in creating specialised

*This paper was initially inspired by the Kaggle competition Google – AI Assistants for Data Tasks with Gemma.

language models for niche domains, but further research is needed to refine the methodology and address the identified challenges. We outline several directions for future work, including scaling up the training set, investigating the impact of answer length on model performance, and conducting more in-depth analyses of answer superiority. The rest of this paper is organised as follows: Section 2 reviews the related literature; Section 3 describes our data generation process; Section 4 outlines our baseline; Section 5 presents details on the methodology; Section 6 presents results and analysis; and Section 8 outlines future work.

2 Literature Review

Recent literature has explored the potential of fine-tuning smaller language models for specific tasks, aiming to create cost-effective and computationally efficient alternatives to large, general-purpose models. Zhao et al. [2024] investigate the effectiveness of Low Rank Adaptation (LoRA) for fine-tuning Language Models (LMs) across a wide range of tasks and models. They explore fine-tuning Gemma-2b-it, among 9 other models, across a broad range of tasks and domains, ranging from grade school math problems to disease recognition and coding tasks in multiple languages. However, for knowledge and reasoning-based tasks, their study is limited to multiple-choice or yes/no questions. Our project aims to extend this by focusing on fine-tuning for specific tasks that involve a combination of knowledge and reasoning, with an emphasis on generating open questions that are more relevant to scientific applications.

Taori et al. [2023] use standard fine-tuning techniques on the 7B and 13B parameter versions of the LLaMA model to fine-tune style, using a synthetic dataset generated by GPT-3.5. Their approach is similar to ours in the sense that it uses synthetic training data produced by an LLM, but it differs in terms of not requiring the model to acquire knowledge from the fine-tuning stage, focusing only on style.

In addition, recent literature has emphasised the feasibility of deploying smaller models on specific domains without compromising accuracy. These models are typically trained on synthetic datasets and fine-tuned versions of LLaMA foundation models, targeting various knowledge domains, such as chemistry McNaughton et al. [2024], medicine Wu et al. [2023], arithmetic Liu and Low [2023], and time series Rasul et al. [2023].

Our work extends the existing literature by concentrating on the under-explored field of data science, using high-quality knowledge sources, and emphasising the use of compact, open-source language models that can be fine-tuned and deployed on consumer-grade hardware. We focus on the Gemma-2b-it model, one of the few 2B models available as open-source – alongside Phi-2b from Microsoft, GPT-2 from OpenAI, and Stable LM 2 from Stability AI. While the literature has mostly focused on 7B models, which require more computational resources, we investigate whether narrowing the subject and providing knowledge from reliable sources can aid performance gains in the smallest available models.

3 Data

We have generated a dataset containing 4,544 question-answer pairs based on the following books: "Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani; "Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman; "Introduction to Probability" by Joseph K. Blitzstein and Jessica Hwang; and "Computer Age Statistical Inference: Algorithms, Evidence, and Data Science" by Bradley Efron and Trevor Hastie. To ensure the quality of the Q&A pairs, we used Claude-3 Opus with a carefully designed prompt.

The prompt provides clear guidelines for creating diverse question types, covering a wide range of topics and catering to different levels of understanding. It also emphasises the generation of detailed, self-contained answers that include relevant examples and analogies to clarify complex ideas.

We split the data into 4,054 training examples and 470 test examples.

4 Baseline

As a baseline, we use the raw Gemma-2B-it model to answer the same questions from the held-out test set. This baseline lacks the domain-specific fine-tuning that our proposed approach incorporates.

We use the ROUGE-L metric to compare the answers generated by the baseline model to the ground truth answers in the test set. On average, the ROUGE-L score for the baseline model is 0.28, indicating a somewhat low level of similarity between the generated answers and the ground truth.

The baseline model's responses generally contain several issues, such as typos and grammatical errors, repetition of certain points, abrupt and incomplete endings, lack of clarity in some explanations, and absence of concrete examples to illustrate concepts. These limitations highlight the need for domain-specific fine-tuning to improve the quality, coherence, and relevance of the generated answers.

5 Main approach

To efficiently adapt the Gemma-2b-it model to the data science domain, we use LoRA for fine-tuning – a parameter-efficient method that adds a small number of trainable parameters to the pre-trained model, reducing the memory footprint and computational requirements compared to traditional fine-tuning approaches.

We fine-tune the Gemma-2b-it model using the following hyper-parameters:

Hyper-parameter	FT, FT-2x	FT-2x-h
LoRA Rank	4	5
Trainable params	1,363,968	1,704,960
Learning rate	5e-5	5e-5
Epochs	1	1
Batch size	1	1
Max length	512	614
Weight decay	0.01	0.01

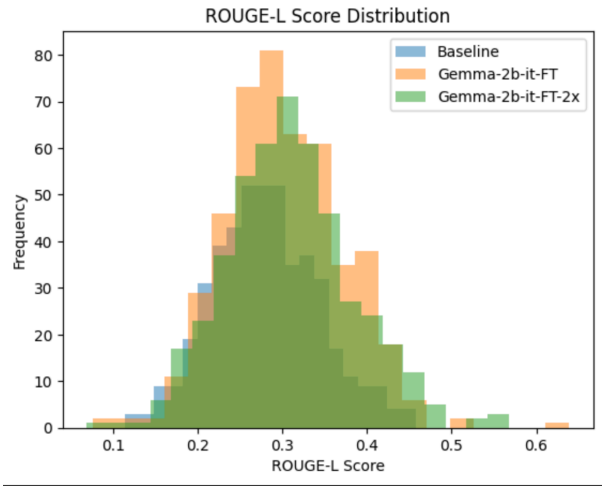
Table 1: Hyper-parameters for fine-tuning Gemma-2b-it.

The process is performed on a T4 GPU in Google Colab with 15GB of RAM[1], using Keras with a JAX backend and standard code from Google.

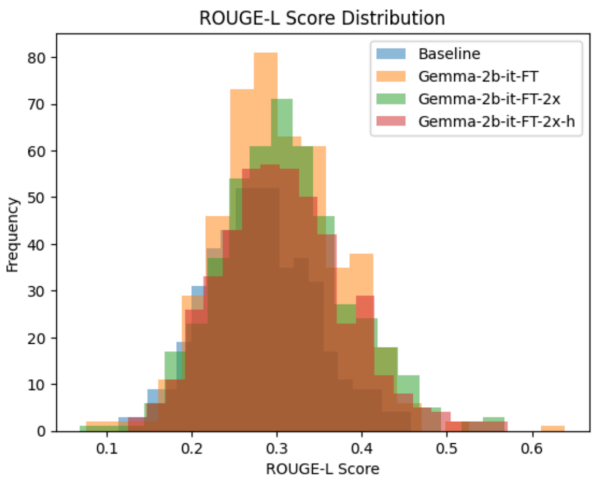
The input to the model consists of question-answer pairs generated from the selected data science textbooks, with the questions serving as the input and the corresponding answers as the target output.

During fine-tuning, the model learns to adapt its knowledge to the specific domain by updating the LoRA parameters while keeping the pre-trained weights frozen. This allows the model to retain its general language understanding capabilities while specialising in the data science domain.

6 Results & Analysis



(a) Baseline vs FT vs FT-2x



(b) Baseline vs FT vs FT-2x vs FT-2x-h

Figure 1: Comparison of ROUGE-L scores.

[1]We were able to process roughly 2,000 examples per hour.

Metric	Baseline	FT	FT-2x	FT-2x-h
Average ROUGE-L	0.28	0.30	0.31	0.31
Median ROUGE-L	0.28	0.30	0.30	0.30
Standard deviation ROUGE-L	0.06	0.07	0.07	0.07

Table 2: Moments of the Distribution of ROUGE-L scores.

We initially fine-tuned the model twice. The first model, Gemma-2b-it-FT (FT), was trained with 1,877 examples, while the second model, Gemma-2b-it-FT-2x (FT-2x), was trained with 4,054 examples. Both models were trained with the parameters displayed in Table 1. With FT-2x, we aimed to evaluate the effect of the number of training examples on model performance.

Figure 1a shows improved performance of the fine-tuned Gemma-2b-it models compared to the baseline, indicating higher similarity between generated and ground truth answers. However, the median score difference is only 7%, and the fine-tuned models have an 11% wider standard deviation. We would have expected that twice as much data would generate higher improvements than the observed 3% improvement on the mean, due to its fatter right tail of the score distribution.

ROUGE-L measures similarity between the fine-tuned model's answer and the test data, but low similarity may not be a bad outcome since high similarity may indicate overfitting. To better assess improvement, we used GPT-4, an independent state-of-the-art model, to evaluate answers from the three models using a rubric that captures comprehensiveness, accuracy, clarity, self-containment, and logical structure. The scores within each item can be 0, 0.5, or 1, so a perfect answer would receive a maximum of 5.

Interestingly, the FT model wins 41% of the test set questions, while the FT-2x model wins 35%. They are evaluated as tied[2] with equivalent good answers superior to the baseline answer in 10% of the questions. This suggests that while increasing the training data does lead to some improvements, the relationship between the amount of training data and model performance is not strictly linear. The evaluation using GPT-4 and a comprehensive rubric reveals that the model trained with fewer examples can still outperform the one trained with more data in certain aspects. This highlights the importance of using multiple evaluation metrics and considering qualitative factors when assessing the gains achieved by fine-tuning a model.

If the models were equivalent, we should expect a higher percentage of ties. This result indicates that the two models may perform better on different types of questions. If that's the case, we could develop a classification model that sends different questions to different models, potentially achieving a total of 86% superior answers to the baseline model.

Overall, based on the scores and upon closer inspection, the fine-tuned models are significantly superior to the baseline.

[2]Ties between the baseline model and the others were solved by giving the win to the baseline model.

To experiment with hyper-parameters, we fine-tuned a third model, FT-2x-h, with the same training data used in FT-2x but increased the rank from 4 to 5, resulting in 25% more trained parameters (341,000). Additionally, we increased the sequence length by 20% to better handle the longer answer lengths in the training data. For a comparison, see Table 1.

With hyper-parameter changes, Figure 1b shows that we seem to continue to move the distribution of scores to the right, though the distribution moments remain mostly unchanged (Table 2). This indicates that with some more calibration, we could potentially achieve good results with just one well-calibrated model during tuning, as opposed to the suggested ensemble of models, which could be more computationally efficient. Due to time constraints, we have unfortunately not been able to include the answers from this model in the GPT-4 evaluation, which we leave as a future exercise and explore punctually in the next section.

7 Error Analysis

To gain a more objective understanding of the outputs from the different models, let us examine their answers to the same question below.

```
Question: How do the coefficient estimates differ when using a large
   ↪  value of lambda in ridge regression compared to a small
   ↪ lambda value?"


Test set: With a large value of lambda, the L2 regularization
   ↪ penalty is high, so the ridge regression coefficient estimates
   ↪  will have a much smaller L2 norm (Euclidean length). The
   ↪ coefficients are shrunken significantly towards zero.
   ↪ Conversely, a small lambda applies little regularization,
   ↪ allowing the coefficients to take on larger values to fit the
   ↪ training data more closely. The L2 norm of the coefficients
   ↪ will be much larger with small lambda.

Gemma-2b-it: The larger is lambda, the larger will be the
   ↪ coefficient of the variables.\nAs a result, the coefficient
   ↪ will be closer to zero and the model will be less accurate.

Gemma-2b-it-FT: When using a large lambda value, the coefficients
   ↪ are shrunk to zero, while a small value of lambda shrinks them
   ↪  towards 0 with a much slower rate. In other words, a large
   ↪ value shrinks the coefficients more strongly than a small
   ↪ value. This can lead to very different estimates of the
   ↪ coefficients and the regression line. A larger lambda can help
   ↪  prevent overfitting, but it may also lead to less accurate
   ↪ estimates, as coefficients can be driven too close to zero.
```

```
Gemma-2b-it-FT-2x: When using a large lambda value in ridge
   ↪ regression, the coefficients become much smaller in magnitude,
   ↪  and the signs of the coefficients may even switch. This is
   ↪ because the ridge penalty becomes more dominant, and the model
   ↪  becomes less sensitive to the individual coefficient
   ↪ estimates. As a result, some coefficients may become zero, and
   ↪  those with non-zero estimates may be close to each other in
   ↪ magnitude.\n\nOn the other hand, when using a small lambda
   ↪ value in ridge regression, the coefficients tend to remain
   ↪ close to the original estimates. This is because the ridge
   ↪ penalty is less dominant, and the model is not as sensitive to
   ↪  the individual coefficient estimates, allowing them to remain
   ↪  relatively unchanged. The model is more robust and flexible,
   ↪ allowing it to capture a wider variety of relationships
   ↪ between the predictors and the response.

Gemma-2b-it-FT-2x-h: When the coefficient estimates are large and
   ↪ the ridge parameter \u03bb is large, the model will have a
   ↪ large amount of regularization. This results in coefficients
   ↪ with smaller absolute magnitude, but still non-zero. In other
   ↪ words, the model is more likely to select coefficients close
   ↪ to zero, even for variables with small absolute coefficients
   ↪ in the unregularized model. As \u03bb increases, the penalty
   ↪ increases and the model becomes more regularized, leading to
   ↪ coefficients with smaller magnitudes. The tradeoff between
   ↪ model complexity and regularization is evident in the
   ↪ coefficients.",
```

GPT-4 selected Gemma-2b-it-FT's answer, stating that "the flow is a bit choppy" in Gemma-2b-it-FT-2x. However, I believe the latter provides a more comprehensive answer and even preempts follow-up questions by mentioning that ridge regression may switch the signs of coefficients, which, upon inspection, appears to be accurate. Gemma-2b-it-FT-2x-h is generally correct and captures the main ideas, but it could benefit from some clarification and more precise language to avoid potential misinterpretations. When asked to choose, Claude-Opus picks the test set response as the best answer but selects Gemma-2b-it-FT-2x as the second-best option.

This highlights the subjectivity of the choices and fluctuations in model feedback, making it less reliable and not a complete replacement for human feedback. As a follow-up, it would be beneficial to have ratings from impartial experts to assess the answers. However, the model rating serves as an excellent initial filter for the particularly poor answers, saving experts from having to evaluate too many choices, as the raw model's answer is clearly subpar.

The results presented in Section 6 may be affected by several factors, including but not limited to:

- The method used to generate the second half of the training data: Since the questions were based on the same books and only varied in the amount of information per batch

and the number of questions prompted to be generated, this may have led to overly similar questions that do not add new information to the model.

- The length of answers in the training data: Typically, the training data contains long answers, so length restrictions during the training process may curtail the potential for learning. Shorter and more objective answers would probably be preferred.
- The effectiveness of ROUGE-L in measuring performance: As the comparison is made based on sentence length, the length restrictions in response data may have affected the effectiveness of ROUGE-L. Exploring alternative automated metrics may be beneficial.
- The lack of in-depth analysis of answer superiority: We did not further explore the degree to which one answer is superior to another by evaluating the scores for each rubric item in the GPT-4 evaluation. We could potentially learn more by assigning weights to the items deemed more important, for example, accuracy of the answers.
- Limitations in computational resources: The exploration of other hyper-parameters and their effects on model quality was hindered by computational resource constraints. Generating heuristics on this topic would be valuable to serve as a roadmap for users with limited computing resources, which was the goal of the paper.

8 Future Work

- Scale up the training set to investigate the relationship between the amount of new information introduced during fine-tuning and model performance. Ensure that the augmented dataset contains diverse and informative questions to provide meaningful insights into the model's learning capabilities.
- Explore the impact of answer length on model performance by fine-tuning the model with shorter, more concise answers. This may help the model learn more effectively and generate more objective responses.
- Investigate alternative automated metrics that are better suited for evaluating the performance of models trained on datasets with varying answer lengths.
- Conduct a more in-depth analysis of answer superiority by evaluating the scores for each rubric item in the GPT-4 evaluation. Assign weights to the items based on their importance and analyze the results to gain a better understanding of the model's strengths and weaknesses.
- Fine-tune hyper-parameters, especially increasing LoRA's rank, the number of trainable parameters, and the layers they apply to. Explore the impact of these adjustments on model performance and generate heuristics to guide users with limited computational resources.
- Obtain domain experts feedback on the evaluation of answers.

9 Code

This project's repo can be found on Github.

References

Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. 5 2023. URL http://arxiv.org/abs/2305.14201.

Andrew D. McNaughton, Gautham Ramalaxmi, Agustin Kruel, Carter R. Knutson, Rohith A. Varikoti, and Neeraj Kumar. Cactus: Chemistry agent connecting tool-usage to science. 5 2024. URL http://arxiv.org/abs/2405.00972.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting. 10 2023. URL http://arxiv.org/abs/2310.08278.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-llama: Towards building open-source language models for medicine. 4 2023. URL http://arxiv.org/abs/2304.14454.

Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. Lora land: 310 fine-tuned llms that rival gpt-4, a technical report. 4 2024. URL http://arxiv.org/abs/2405.00732.