Fine-Tuning Large Language Models to Generate Solutions for NLP Research Problems

Lakshmi Chandrika Yarlaggadda (lvy5215)

April 25, 2025

1 Abstract

Large Language Models have demonstrated remarkable capabilities in language understanding and generation tasks. This project focuses on fine-tuning open-source LLMs to generate solutions for research problems specifically within the field of Natural Language Processing. In this project first, we extract abstracts from NLP-related papers and use the LLaMA model to generate problem-solution pairs dataset. These pairs are used to fine-tune open source LLMs to generate approaches for solving NLP research problems.

2 Related Work

Recent research has shown that fine-tuning large language models (LLMs) on specific instruction-following datasets can make them much more competent at specialized tasks. Several important projects shaped this idea and were relevant to our project:

2.1 Alpaca: Fine-Tuning LLaMA with Instruction Data

Stanford's Alpaca project [1] showed that one can significantly improve the performance of a baseline model like LLaMA-7B with a small instruction-following dataset. They generated 52,000 instruction-response pairs by employing OpenAI models and fine-tuning LLaMA and built a model that could perform human instructions correctly with much less resources. This encouraged our suggestion to use synthetic datasets for fine-tuning.

2.2 Self-Instruct: Teaching Models with Their Own Outputs

Self-Instruct [2] introduced the idea of starting with a few examples and asking a pre-trained model to generate a great many more instructions and responses on its own. This lowered the requirement for human-labeled data. We adopted the same idea when we employed LLaMA to create problem-solution pairs from abstracts.

2.3 Flan-T5: Scaling Instruction Tuning

Flan-T5 [3] is Google's work where T5 models were optimized on a highly diverse set of tasks through instruction tuning. It showed that scaling instruction sets improves generalization to new, unseen tasks. This motivated us to tackle our problem of utilizing many different types of problems in order to make the model more flexible.

2.4 Dolly 2.0: Open Source Instruction Models

Dolly 2.0 by Databricks [4] created an open dataset and fine-tuned their own model without relying on proprietary APIs. It showed that it is possible to build strong models using solely open and public resources, which aligns well with our vision of creating an open and reproducible fine-tuning pipeline.

2.5 Vicuna: Fine-Tuning for Dialogue and Assistance

Vicuna [5] is an improved version of LLaMA that is created for open-ended dialogue. Though it is meant for dialogue, as opposed to solving problems, it shows how LLaMA models can be adapted to perform complex multi-turn instructions, which is applicable when generating structured solutions to research in our project.

3 Problem Statement and Dataset Curation

3.1 Problem Statement

In this project, we want to fine-tune open-source LLMs to generate approach/solutions to NLP research problems. The model should be able to read a problem description and provide a coherent solution, similar to a human researcher outlining an approach.

3.2 Dataset Curation

3.2.1 Abstract Collection

Abstracts were collected from arXiv papers in the `cs.CL` (Computation and Language) category, focusing specifically on NLP research. Each abstract was parsed to identify the problem and the corresponding proposed solution.

3.2.2 LLM-Based Data Generation

A pre-trained LLaMA model was used to transform each abstract into a question-answer pair. The following prompt structure was used:

> You are a helpful assistant that extracts structured information from scientific research abstracts. Your task is to identify the main PROBLEM the research addresses and summarize the SOLUTION proposed.Always include technical details, methods, or

techniques used in the SOLUTION if available.Given the following abstract, extract and label the PROBLEM and the SOLUTION: Abstract: text[:2000]

For example,

Problem: Traditional RAG frameworks are ineffective with visually-rich documents like PDFs containing charts and tables. These systems rely on parsing text and often miss important visual information.

Solution: The VDocRAG framework addresses this by processing entire documents as images, using vision-language models with novel pre-training tasks to encode visual and textual features. The OpenDocVQA dataset is introduced for evaluating such models.

3.2.3 Cleaning

Generated pairs were then manually checked to see if the problem and the solution are related and coherent atleast to certain extent. Pairs without any structure or sense are dropped.

3.2.4 Preprocessing

- Removal of special characters, if any.
- Truncation or padding of responses to fit model input limits.
- Any instruction-response pairs with empty or extremely short fields were filtered out to maintain dataset quality.

4 Model Architecture

The architecture for this project involves a two-stage pipeline: (1) generating a synthetic dataset from research abstracts using large pre-trained language models, and (2) fine-tuning smaller open source models on this dataset to generate structured solutions to NLP research problems.

4.1 Data Generator Models: LLaMA and BART

To generate high-quality problem-solution pairs from NLP-related research abstracts, two pre-trained large language models were used: LLaMA-7B and BART-Large.

4.1.1 LLaMA

LLaMA (Large Language Model Meta AI) [7] is a family of transformer models built by Meta AI. It has a decoder-only architecture, which means it generates text by predicting tokens sequentially. We used the 7-billion parameter variant of LLaMA in this project to help generate the training data. We exposed it to research paper abstracts and asked it to break them down into two parts: a simple description of the research problem and a systematic overview of the proposed solution.

LLaMA was able to produce good-quality question-answer pairs, which we then used to fine-tune small models.

4.1.2 BART

BART [8] (Bidirectional and Auto-Regressive Transformers)is a Facebook AI model that integrates two powerful ideas: a bidirectional encoder (e.g., BERT) to understand input well, and an autoregressive decoder (e.g., GPT) to generate fluent output. BART is trained to recover input texts that are corrupted and is very powerful on tasks like summarization and rewriting. We used BART in this project to produce varied problem-solution pairs from research abstracts, with the goal of having more readable and varied outputs.

4.2 Target Models for Fine-Tuning

After constructing the dataset, three open-source models were fine-tuned: T5-Small, T5-Base, and Mistral-7B.

4.2.1 T5-Small

T5-Small [9] is a smaller version of the T5 model, with around 60 million parameters. It is based on an encoder-decoder architecture, where the input text is first encoded into hidden features and then decoded into an output text. We chose T5-Small to experiment with how well a light-weight model would work when trained to generate structured solutions. It was useful in determining the limitations of smaller models in research tasks.

4.2.2 T5-Base

T5-Base is a larger model than T5-Small, consisting of a total of approximately 220 million parameters. It also uses an encoder-decoder paradigm but with higher capacity, i.e., it is able to accept more complex inputs and produce higher quality outputs. We have included T5-Base to see how much the performance improves as we move from a small to a medium-sized model, especially in the tasks which require detailed and accurate responses.

4.2.3 Mistral-7B

Mistral-7B [10] is a much deeper model with 7 billion parameters. It's transformer-based and uses advanced techniques like grouped-query attention and sliding window attention to improve its performance and efficiency. Mistral-7B is said to achieve very high quality in reasoning and instruction-following tasks. We chose it to be our large model of first choice to fine-tune as it already worked well and could generate high-quality, step-by-step solutions as soon as we had trained it on our data.

5 Experiments and Results

5.1 Evaluation Metrics

- **BLEU Score**: Measures n-gram precision between generated and reference texts.
- **ROUGE-1, ROUGE-2, ROUGE-L**: Measures the overlap of unigrams, bigrams, and longest common subsequence respectively.
- **BERTScore (F1)**: Measures semantic similarity between predicted and reference texts using contextual embeddings.

5.2 Dataset Quality Comparison: LLAMA vs BART

The LLaMA-generated dataset is more structured, detailed, and helpful than that of BART. LLaMA neatly marks the problem and solution, with technical steps, methodologies, and framework names, hence being more practical for model training in addressing research problems. BART output merely copies or rephrases the original abstracts without giving a clear identification of problem and solution. Overall, LLaMA provides better-quality data and it has been used for fine-tuning.

5.3 Qualitative Analysis

Table 1 shows predictions from each of the three models on the same research problem. This comparison highlights differences in relevance, completeness, and factual consistency.

5.3.1 Analysis

5.4 Model-wise Observations

- **T5-Small:** Outputs were often redundant, incoherent, and hallucinated words that were not present in the original research. The model was unable to capture the logical structure of solutions appropriately and tended to develop new frameworks or methodologies.
- **T5-Base:** Compared to T5-Small, T5-Base generated somewhat better-structured outputs with some technical terms. However, it fabricated solutions and frameworks that doesnot exist.
- **Mistral-7B:** Mistral-7B always produced the most precise and semantically nearest responses. Although the model occasionally used different wording, the fundamental ideas of the original research answers were correctly caught. The outputs were more extensive, well-coordinated, and stylistically appropriate for research settings.
- **Progression:** T5-Small $\rightarrow$ T5-Base $\rightarrow$ Mistral-7B showed a clear and consistent improvement across all qualitative metrics.
- **Mistral-7B Strengths:** Superior faithfulness, relevance, and fluency make it much more reliable for real-world research assistance.

Research Problem: The problem addressed by the research is the ”deep-rooted one-pass issue” in large language model (LLM) agents, where each generated intermediate thought is plugged into the trajectory regardless of its correctness, leading to irreversible error propagation. **Solution from the paper:** The Generator-Assistant Stepwise Rollback (GA-Rollback) framework utilizes a generator to interact with the environment and an assistant to examine each action. The assistant triggers rollback for incorrect steps. Two additional rollback-specific strategies are also introduced.
T5-Small Prediction: The solution involves a single-pass solution called a reversible error propagation (reversible error propagation) that combines a single-pass solution with a single-pass solution. **Comment**: Repetitive, incoherent, and factually incorrect. Hallucinated terminology.
T5-Base Prediction: The proposed solution is a novel approach that leverages the inverse-reverse-correction approach to generate intermediate thought... **Comment**: Some structure is present, but core concepts are fabricated. The ”inverse-reverse-correction” approach is not part of the target.
Mistral-7B Prediction: The proposed solution is the ”Thoughtful Agent” framework, which introduces a novel agent that dynamically revises intermediate thoughts. It includes a reward function and training strategy to encourage revision of flawed reasoning. **Comment**: Captures the core idea of rollback and dynamic correction. Uses different terminology but remains semantically aligned.

Table 1: Representative model predictions for a single research problem. Mistral-7B shows the highest semantic alignment and coherence.

- **Common Challenges:** All models struggled slightly with replicating very fine-grained technical details unless explicitly guided, suggesting further room for improvement with more fine-grained instruction tuning.

5.5 Quantitative Results

All the three models are compared with the evaluation metrics that have been mentioned earlier.

Metric	T5-Small	T5-Base	Mistral-7B
BLEU Score	0.0447	0.0503	**0.0702**
ROUGE-1	0.2411	0.2522	**0.3180**
ROUGE-2	0.0597	0.0637	**0.0848**
ROUGE-L	0.1838	0.1907	**0.2120**
BERTScore (F1)	0.8325	0.8410	**0.8586**

Table 2: Performance comparison of different models fine-tuned on the NLP research problem dataset.

5.5.1 Analysis

The results show how fine-tuning helps model performance considerably across all tasks. T5-Base performed better than T5-Small, showing the value of using larger pre-trained models for instruc-

tion tuning.

Mistral-7B did best overall, especially in ROUGE-1, ROUGE-2, and BLEU. It shows that it is better at producing precise and relevant answers. The BERTScore F1 also increased significantly, which indicates that Mistral's performance was more semantically close to the target responses compared to other smaller models.

These findings validate the decision to utilize Mistral-7B as an appropriate base model for solving specific research problems in NLP by fine-tuning.

5.6 Hyperparameter Tuning

To further optimize model performance, experiments were conducted by varying batch size, learning rate, and number of epochs. Table 3 summarizes the impact of these hyperparameters on the Mistral-7B model.

Configuration	BLEU	ROUGE-L	BERTScore (F1)
3 epochs, batch size 4, LR 0.001 (baseline)	0.0702	0.2120	0.8586
5 epochs, batch size 4, LR 0.001	0.0755	0.2215	0.8647
3 epochs, batch size 8, LR 0.001	0.0783	0.2280	0.8664
5 epochs, batch size 8, LR 0.001 (best)	**0.0817**	**0.2356**	**0.8691**
5 epochs, batch size 8, LR 0.01	0.0601	0.1982	0.8458

Table 3: Effect of batch size, learning rate, and epochs on Mistral-7B fine-tuning performance.

5.6.1 Analysis

Increasing the batch size from 4 to 8 improved BLEU, ROUGE-L, and BERTScore scores consistently, demonstrating that larger batches made training more stable and improved. Taking more time with the data by increasing training from 3 to 5 epochs also benefited performance, demonstrating that more time with the data enabled the model to learn more.

On the other hand, raising the learning rate from 0.001 to 0.01 worsened the model across all metrics. This shows that the lower learning rate (0.001) is better for stable fine-tuning on a smaller and more specific dataset.

The best results were achieved with 5 epochs, batch size 8, and learning rate 0.001.

5.7 Ablation Study: Mistral Fine-Tuned vs. Mistral-Instruct Fine-Tuned

In order to better understand the impact of starting point selection, we conducted an ablation study where both the base Mistral-7B model and the Mistral-7B-Instruct model were fine-tuned from a single NLP research dataset.

Unusually, we found the fine-tuned Mistral-7B-Instruct model to be performing very slightly better than the fine-tuned Mistral-7B model across most of our evaluation metrics. This suggests that starting from a pre-existing instruction-tuned checkpoint such as Mistral-Instruct has advantages such as better early alignment to task form and more natural response output. The improvements, though marginal, indicate that Instruct models are a good place to start from for domain-specific fine-tuning tasks over vanilla base models.

6 Error Analysis

While the models produced good results overall, several types of mistakes were noticed during evaluation:

- The T5 models sometimes invented fake methods/frameworks names that do not exist or do not correlate to the problem. This made the generated outputs seem more creative but less accurate.
- In many cases, the models left out important parts of the original solutions and gave a high level generic solution instead of a specific one.
- Sometimes. even when the models were able to propose a similar aproach, the wording made it look semantically different.
- For problems requiring multi-step solutions, some models misunderstood key ideas. For example, on a question regarding the GA-Rollback framework (designed to allow an agent to undo erroneous intermediate steps), the T5-Small model incorrectly described it as a r̈eversible single-pass solutionẅith no mention of the rollback. The correct solution is a generator-assistant framework where the assistant is actively monitoring and triggers a rollback if it detects an error. This shows that the model failed to capture the crucial multi-agent rollback logic
- The smaller T5 models had more hallucination and missing details, while Mistral-7B was generally better at sticking to the correct concepts. However, even Mistral would miss very fine-grained technical details unless the input was extremely clear.

Error patterns indicate that while fine-tuning improves alignment and coherence, challenges remain to faithfully capture nuanced scientific content without hallucination.

7 Lessons Learnt

Several important lessons were learnt in the project:

- Creating the dataset with pre-trained models, saved lot of effort and time.
- Clear prompts will help the model in generating better results.
- Problem-solution pairs generated by LLAMA are more coherent and helped the model to learn better than the ones generated by BART.

- Larger models like Mistral-7B gave more coherent and detailed solutions compared to models like T5.
- Hyperparameter tuning is important and it improved the model performance.
- Larger models performed better than smaller models.

8 Challenges

During the project, several challenges were encountered:

- Running large models like Mistral-7B required high computational resources, and memory limitations made fine-tuning slower, especially when using larger batch sizes.
- Generating datasets using LLaMA and BART sometimes produced noisy or incoherent question-answer pairs, requiring manual filtering to ensure high-quality training data.
- Despite fine-tuning, models occasionally mention non-existant methods or missed fine-grained technical details.

9 Future Work

Potential avenues for extending this work include:

- Instead of just using the abstract, compile digests of the papers and to create the question-answer pairs.
- Increase the size of the dataset by considering different conference papers and NLP archives.
- Develop a benchmark dataset to fine tune the models so that they can generate novel approaches on their own.
- Instead of manually checking for quality analysis, implement fact checking modules to check how effective the solutions are.
- Develop RAG based systems that can help the model to have complete knowledge about a domain, so that it can come up with different approaches.
- Explore techniques to fasten the model finetuning process with huge datasets.

10 Conclusion

This project demonstrated that fine-tuning open-source large language models using synthetically generated instruction-response pairs enables the creation of lightweight models capable of providing structured solutions to NLP research problems. From our experiments, Mistral-7B is the most reliable model, achieving the highest scores in BLEU, ROUGE, and BERTScore metrics. Qualitative analysis further validated its superior factfulness, relevance, and completeness compared to

smaller models.

Despite some persisting challenges, particularly around hallucination and fine-grained detail retention, this project shows the feasiblity of improving the performance of the LLMs to generate novel solutions to research problems by finetuning them with the existing problem set.

References

[1] R. Taori, I. Shliazhko, X. Zhang, Y. Dubois, S. Guestrin, P. Liang, and T. Hashimoto, *Stanford Alpaca: An Instruction-Following LLaMA Model*, 2023. Available at: `https://github.com/tatsu-lab/stanford_alpaca`

[2] Y. Wang, J. Kordi, D. Mishra, Y. Xie, S. Xu, H. Chung, et al., *Self-Instruct: Aligning Language Models with Self-Generated Instructions*, arXiv preprint arXiv:2212.10560, 2022.

[3] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, P. Barham, E. Li, X. Wang, M. Dehghani, and others, "Scaling Instruction-Finetuned Language Models," arXiv preprint arXiv:2210.11416, 2022.

[4] Databricks Team, *Dolly 2.0: Democratizing the magic of ChatGPT with open models*, 2023. Available at: `https://www.databricks.com/blog/2023/04/12/dolly-v2-open-instruction-following-llm.html`

[5] Z. Chiang, Z. Xiao, H. Xu, X. Zheng, M. Guo, A. Vaswani, L. Zettlemoyer, and L. Tu, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality," arXiv preprint arXiv:2304.08994, 2023.

[6] T. Kopf, A. Glatzer, Y. Skital, et al., *OpenAssistant Conversations - Democratizing Large Language Model Alignment*, arXiv preprint arXiv:2304.07327, 2023.

[7] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, et al., *LLaMA 2: Open Foundation and Fine-Tuned Chat Models*, arXiv preprint arXiv:2307.09288, 2023.

[8] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. In Proceedings of ACL 2020. arXiv preprint arXiv:1910.13461.

[9] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. Journal of Machine Learning Research, 21(140), 1–67.

[10] T. Jiang, S. Liu, and A. Rashkin, *Mistral 7B: A Sparse Mixture of Experts Model with State-of-the-Art Performance*, Mistral AI, 2023. Available at: `https://mistral.ai/news/announcing-mistral-7b/`