Automated Tailoring of Large Language Models for Industry-Specific Downstream Tasks

Shreya Saxena Siva Prasad Muneeswaran I Advaith Shankar
Varun V Saisubramaniam Gopalakrishnan Vishal Vaddina

Applied Research, Quantiphi

The 17th ACM International Conference on Web Search and Data Mining

Abstract

Foundational Large Language Models (LLMs) are pre-trained generally on huge corpora encompassing broad subjects to become versatile and generalize to future downstream tasks. However, their effectiveness falls short when dealing with tasks that are highly specialized to a specific use case. Even when adopting current prompt engineering techniques like few-shot or Chain-of-Thought reasoning prompts, the required level of results is not yet achievable directly with foundational models alone. The alternative approach is to fine-tune the LLM, but a common challenge is the limited availability of task-specific training data. We Introduce an end-to-end automated framework to tailor a model to specific downstream tasks for an industry where the first step is to generate task-specific custom data from unstructured documents.

Additionally, we also introduce our optimized distributed training pipeline for fine-tuning LLMs on the generated data.
Finally, we will provide an overview of the statistical metrics and customized metrics we employ for assessing the performance of the fine-tuned LLM.

This automated framework alleviates the burden of manual adjustments and streamlines the process to provide a model that is fully customized to suit the unique requirements of any specific business use case.

Introduction

Due to the expensive and time-consuming process for annotating and curating task-specific data from unstructured texts, there is a demand for an automated and comprehensive framework that can create custom data tailored to downstream tasks effortlessly by just providing the task description and, then fine-tuning the model on this data to enhance its performance on the downstream task.

Challenges in Fine-Tuning

- Foundational LLMs are usually pre-trained on generic tasks and domain
- Using LLMs like GPT, PALM are expensive and might pose privacy issues.
- Prompt engineering is required for specific use-cases might not give favourable results
- Primarily input is available in the form of unstructured texts, documents, PDFs
- Lack of task specific data to finetune model for a specific domain-based or any downstream tasks

Solutions

- Finetune a smaller LLM for domain adaptation, improved task performance and fine-grained control
- Generate high quality task specific data to tailor any Large Language Model

Auto-train Pipeline

End-to-end automated framework: Auto-Train

- Generate custom/task specific data
- Finetune LLM on huge corpus of unstructured data
- Evaluate the performance of the fine-tuned model

Input Data (Unstructured Texts) → Data Generation Pipeline → Distributed Training Pipeline → Evaluation → Inference

Data Generation

There exists a challenge of limited task-specific data availability for fine-tuning an LLM on downstream tasks. A common cause of this challenge is the scarcity of data due to cost and privacy concerns in the industry setting such as medical data and internal financial reports. We address this with our data generation pipeline. Automatic Prompt Generation pipeline first generates pertinent prompts and few-shot examples based on task-specific descriptions provided by the user. We design our method carefully to ensure that these prompts can generate not only relevant data but also encompass some grounding and edge cases, such as unanswerable questions. This approach helps the model establish a strong contextual understanding in line with the provided task-specific data.

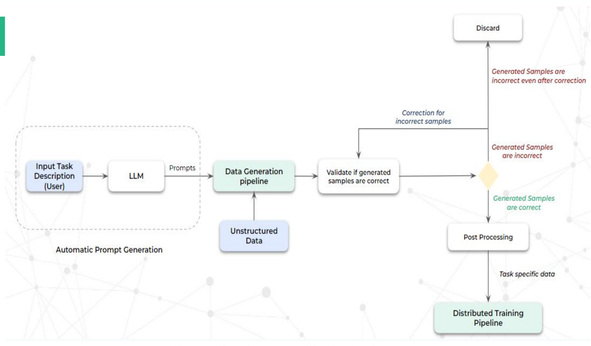

Figure 1: Data Generation Pipeline Architecture

The generated samples are validated by a powerful LLM. If the samples receive a score below a specified threshold, they are reprocessed and reintroduced into the data generation pipeline with feedback. If the samples remain unsatisfactory, they are discarded, while the suitable ones undergo post processing before being incorporated into the training dataset.

Context- Project Report
An American insurance company providing financial protection to more than 50 million customers worldwide. The need for manual review of unlabeled and semistructured documents.....
ABC built a Claim Adjudication Platform to extract, classify, annotate and index the documents submitted for proof of loss and other relevant information....

Automatic Prompt Generation

Summary_prompt:
You are a summarization assistant. Your task is to generate a summary of the given context. Extract the key points, main ideas, and essential information while ensuring the summary remains clear and informative. Make sure the summary is concise and coherent. Avoid including unnecessary details or personal opinions...

Technologies_used_prompt:
You are an text extraction assistant. Your task is to identify and list the technologies used in a given project report. Focus on extracting relevant information related to software, hardware, tools, and frameworks employed in the project. Provide a comprehensive yet concise list of the technologies mentioned in the report. Avoid including details unrelated to the project's technological aspects and refrain from adding personal opinions. Ensure the extracted information is accurate and covers the breadth of technologies discussed in the report.

Data Generation

Summary:
An American insurance company faced challenges in manually reviewing unlabeled and semi structured documents. To address this, ABC developed a Claim Adjudication Platform that automates the extraction,...

Technologies used:
BigQuery, TensorFlow, Cloud Storage

Figure 2: Example data generation for a custom task where input is project reports and the desired output is the summary of the project and the technologies used in that project.

Fine-Tuning

Fine-tuning is essential for the model to adapt to different domains and perform well on specific downstream tasks. The two primary methods of fine-tuning are:- i) Full fine-tuning, where all the model layers are updated while training, and ii) Parameter-efficient fine-tuning (PEFT) [1], where only a subset of model parameters are updated.
We have introduced a framework in our pipeline that allows the user to select between full fine-tuning or LoRA. The users can introduce replay data to avoid catastrophic forgetting [2,3] while full fine-tuning. We have implemented distributed training pipeline using Deepspeed [4] for faster and memory efficient fine-tuning of LLM. We also show-case real-time monitoring and and insights into the fine-tuning model and the ability to configure training settings. This includes setting distributed training strategies, CPU offloading, quantized training, gradient checkpointing, flash attention, number of epochs, etc

- Flash Attention
- Distributed Training ZeRO Strategies
- Mixed Precision Training
- 4-bit and 8-bit quantized training
- CPU offloading
- Gradient Checkpointing
- Gradient Accumulation

Evaluation

Evaluation is pivotal in ensuring that the model not only performs well but also adheres to various critical criteria such as factual accuracy, helpfulness, fairness, etc. To measure these important attributes, we employ a multi-faceted approach, where firstly we utilize statistical-based evaluation methods like bert_score, ROUGE, BLEU on the test-set. Recognizing that traditional evaluation methods may not be fully suited to evaluate generative responses from LLMs, hence, we have developed custom metrics as well. These metrics allow us to thoroughly assess the model's output, grading its responses and verifying their factual correctness.

- Traditional Evaluation Metrics
  - *ROUGE*
  - *BLEU*
  - *BERT score*
- Custom LLM-based Metric
  - *AUTO GRADER*

Next Steps and Future Work

- Adding Multimodality into data generation pipeline
- Automatically setting training configurations in the fine-tuning pipeline

References

[1] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In International Conference on Ma- chine Learning. PMLR, 2790–2799.

[2] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. arXiv preprint arXiv:2004.12651 (2020).

[3] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. arXiv preprint arXiv:2308.08747 (2023).

[4] Rajbhandari, Samyam, et al. "Zero: Memory optimizations toward training trillion parameter models." SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2020.