Chapter 35
Fine-Tuned Large Language Model for Banking

Suresh Limkar, Soham Nale, Shreyash Darade, Ashwin Chaudhari, and Atharva Taras

Abstract A new, improved computer program made specifically for banking helps in solving problems like inadequate guidance and makes it convenient for people who cannot visit banks for some reason. This helps people answer questions faster and makes it easier for bank workers. Making the program better involved picking the right model and making a special set of data. After making these improvements, the llama2_sharded model is really good at answering banking questions, doing better than others by 2.67%. This shows how important it is to give programs lots of practice with different information, proven by reaching a low error rate after many tries.

35.1 Introduction

Banking is really important in people's daily lives, being a key part of how money works. It does lots of things like keeping money safe, lending money for different reasons, and helping with all kinds of financial needs. Almost everyone has a bank account, showing how much banking matters.

When people deal with banking, they often face problems like not getting enough help from bank workers or having to go to a branch for small questions. This takes up a lot of time for people. To fix these problems smoothly, we suggest using a smart computer program made just for banking. This program can help people understand and solve their questions or problems quickly and effectively.

This new idea directly deals with the big problem of people relying too much on bank staff for small questions. By cutting down on the time needed and giving people the power to solve their problems from anywhere, it not only makes people more independent but also makes things easier for bank workers.

S. Limkar (✉)
Department of Computer Science and Engineering, Central University of Jammu, Jammu, Jammu and Kashmir, India
e-mail: sureshlimkar@gmail.com

S. Nale · S. Darade · A. Chaudhari · A. Taras
Department of Artificial Intelligence and Data Science, AISSMS Institute of Information Technology, Pune, India

V. Bhateja et al. (eds.), *Intelligent Data Engineering and Analytics*, Smart Innovation, Systems and Technologies 420, https://doi.org/10.1007/978-981-96-0139-4_35

35.1.1 Artificial Intelligence and Generative AI

Artificial intelligence (AI) is really important in our fast-changing world. It's used for lots of things, like making factories work by themselves and helping computer programmers write code faster. People see AI in different ways. Some think of it as a tool that reduces how much humans have to work, making things happen quicker and better. Others see AI as a system that looks at information and learns from it, using this knowledge to do different tasks [1]. In simpler words, AI is a human-made intelligence which is exhibited by machines in order to assist mankind in various tasks and applications, enhancing overall efficiency and reducing time and cost required [2].

Generative AI utilizes algorithms that possess the ability to produce diverse content types, such as audio, code, images, text, simulations, and videos [3]. In recent years, AI has made a comeback, highlighted by tools like Chat Generative Pre-trained Transformer (ChatGPT). Since its introduction in 2022, generative AI has become a big focus in research. ChatGPT, a type of generative pre-trained transformer, uses deep learning methods to train on a lot of general data. This helps it create different kinds of outputs [4]. Generative AI can create all sorts of content like text, audio, pictures, videos, and even 3D models. Some examples include ChatGPT for text, Midjourney for images, and DeepBrain for videos. These models can work together using text-to-image generation.models [5].

35.1.2 Large Language Models (LLMs)

Language serves as a crucial tool in enabling communication and self-expression among humans and in their interactions with machines. More and more, we need models that can handle complex language tasks like translation, summarization, information retrieval, and conversational interactions. Lately, there have been big improvements in language models, mostly because of transformer architectures [6]. Improvements in technology have brought about a big change, allowing the creation of large language models (LLMs) that can do tasks almost as well as humans [7]. These advanced artificial intelligence systems, known as LLMs, have emerged as capable entities in understanding and creating text that makes sense, showing they can handle different tasks well [8].

Early attempts at creating large language models (LLMs), like T5 and mT5, used transfer learning methods until GPT-3 came along. GPT-3 showed that LLMs could handle different tasks without needing specific training for each one. While LLMs can answer questions about tasks well when given examples, they don't do as well without any training at all. To fix this, we can train LLMs with specific instructions for tasks, making them better at handling new tasks they haven't seen before. This makes them work better without training and makes sure they do what users want them to do [9]. Apart from being better at understanding different topics and adapting to new

areas, large language models (LLMs) can do things like thinking, planning, making decisions, learning as they go, and answering questions in zero-shot settings. They can do these things due to their massive scale, even though they weren't specifically taught these skills during training [10]. These skills have made large language models (LLMs) popular for many different uses, like doing tasks involving different types of information, working with robots, manipulating tools, answering questions, and acting on their own. People have suggested different ways to make them better at these tasks through training that focuses on specific tasks [11].

Large language models (LLMs) can do lots of different tasks almost as well as humans, but they take a long time to train and to give answers, need powerful computers, and cost a lot to run. Because of these needs, not many people use them, so there's a chance to make better designs. People have looked into ways to use LLMs more efficiently, like tuning them to use fewer parameters. Li and Liang [12], pruning, quantization, knowledge distillation, and context length interpolation among others [13].

Large language models (LLMs) such as ChatGPT are changing industries in many ways with their various uses. In business, they make marketing strategies, sales operations, risk assessment, and human resources management easier by helping with communication and analyzing data [14]. In education, ChatGPT plays a crucial role as an assistant, assisting students in information retrieval and language learning while helping educators in lesson planning and feedback [15]. LLMs are greatly beneficial in healthcare, used for supporting clinical diagnoses, providing telehealth services, and educating about health. This leads to better care and outcomes for patients [16]. LLMs also help create content by making personalized ads, showing how versatile and effective they are in making content that people find interesting. Their influence in these areas shows how they can change things a lot, promising a future full of new ideas and better ways of doing things around the world [17].

35.2 Literature Survey

New methods for fine-tuning language models have been suggested, like the pure tuning, safe testing (PTST) approach. This method focuses on safety by separating the fine-tuning part from safety concerns. It doesn't include safety prompts while fine-tuning, but adds them during actual use. The main goal is to solve the problem of safety issues after fine-tuning while still keeping improvements in how well the model does tasks later [18]. Moreover, the three-step fine-tuning for BERT optimization focuses on improving the BERT model for text classification tasks using a detailed three-stage fine-tuning process. This process includes additional pre-training on domain-specific data, optional multi-task learning, and task-specific fine-tuning. Through thorough experimentation, this approach aims to find the best ways to optimize the model. [19]. Moreover, the fine-tuning Llama 2 models for safety and helpfulness approach combines various techniques, such as supervised safety fine-tuning, safety RLHF, and safety context distillation. By using human preference data,

the system trains a reward model, enabling automated preference decisions based on feedback from annotators [20].

The PTST strategy offers advantages in maintaining safety and practical alignment, but further research is needed due to limited understanding. Improved algorithm design is crucial but hindered by high computational costs. Another research direction focuses on optimizing BERT for text classification, but this entails resource-intensive experiments and risks overfitting. Customization is necessary for optimal outcomes across different datasets and tasks. A separate system for Llama 2 models shows improved safety, helpfulness, and scalability but faces challenges such as complex training processes and ethical considerations. Overall, these approaches contribute insights and advancements in fine-tuning language models, each with unique strengths and challenges in safety, effectiveness, and ethical deployment. Current research primarily focuses on text classification, indicating Llama 2 as a promising candidate for secure LLMs in text-based applications due to its emphasis on safety and fine-tuning strategies.

35.3 Proposed System

Fine-tuning a large language model (LLM) for specialized natural language understanding tasks involves various approaches, each with its advantages and considerations. These tasks cover a wide range, including textual entailment, question answering, semantic similarity assessment, and document classification. While large unlabeled text corpora are readily available, the scarcity of labeled data for these specific tasks presents a significant challenge for achieving optimal performance with discriminatively trained models.

The approach used here is based on the fundamental principle of initiating a language model through generative pre-training on a diverse, unlabeled text corpus, followed by a systematic process of discriminative fine-tuning for each specific task. This methodology aims to optimize performance across a range of tasks while preserving the model's structural integrity with minimal architectural changes. It relies on the idea that large language models can capture the complexities of language, facilitating the efficient transfer of knowledge to domain-specific tasks.

Steps to Fine-Tune a Model:

Fine-tuning a language model (LLM) involves several key steps, from defining the task to deploying the final model.

Here Are the General Steps for Fine-Tuning an LLM:

1. **Define Task and Dataset**: Clearly outline the task you want the LLM to perform well on. Collect and prepare a dataset that suits your task, ensuring it's relevant and well-organized for training and evaluation.

2. **Select LLM Architecture**: Choose the most appropriate architecture for your task from available pre-trained LLMs like Misteral.AI, LLAMA2, GPT, and their variations. This decision significantly impacts the fine-tuning process.
3. **Update Model Weights**: Initialize the LLM with pre-trained weights and embeddings, then fine-tune the model using your dataset to adapt it to your specific task. This involves training the model to learn task-specific features.
4. **Select Hyperparameters**: Determine and adjust hyperparameters like learning rates, batch sizes, and training epochs to fine-tune model performance. Experiment with various settings to optimize results, which may require iterative adjustments.
5. **Evaluate Model**: Thoroughly assess your model's performance post fine-tuning using relevant evaluation metrics. Utilize validation and test datasets to measure accuracy, precision, recall, and other pertinent metrics to ensure the model meets your task requirements.
6. **Deploy Model**: Once satisfied with the model's performance, integrate it into real-world applications. Deploy the model in user interfaces, web services, or other platforms where it can interact with users or process data in line with your defined task [21].

There Are Three Main Approaches to Fine-Tuning:

1. **Feature-Based Approach**: This method involves using a pre-trained large language model to generate output embeddings for the training dataset. These embeddings become input features for training a classification model, such as logistic regression, random forest, or XGBoost. Linear classifiers like logistic regression usually yield the best results with this approach.
2. **Fine-tuning I—Updating The Output Layers**: An extension of the feature-based method, this approach keeps the parameters of the pre-trained model intact while training new output layers. It's similar to training a logistic regression classifier or a small multi-layer perceptron on the embedded features. While it offers comparable performance and speed to the feature-based approach, it may be more suitable for scenarios preferring pre-computed and stored embedded features.
3. **Fine-tuning II—Updating All Layers**: The most comprehensive approach involves updating all layers of the pre-trained large language model. While early work on models like BERT suggested fine-tuning only the output layer can achieve comparable performance, optimizing modeling performance often requires updating all layers. This comprehensive fine-tuning is more computationally expensive due to the increased number of parameters but typically yields superior results [3] (Fig. 35.1).

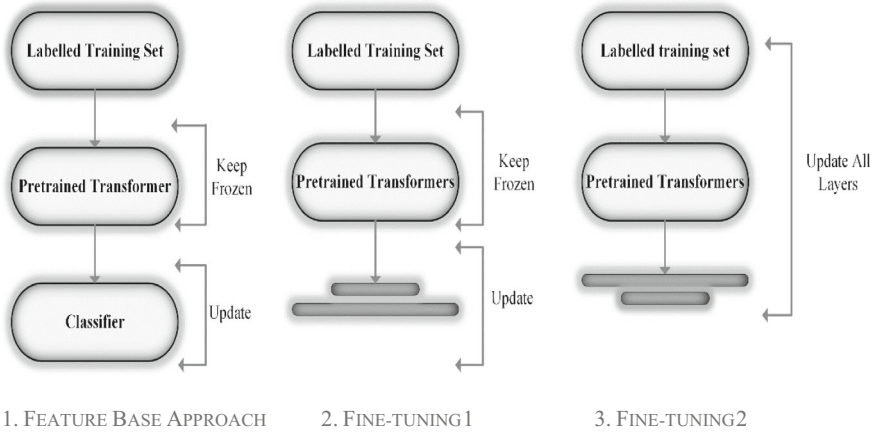

Fig. 35.1 Architecture for various fine-tuning process [3]

35.4 Results and Comparison

In our study, we trained a Llama2 large language model and compared the efficiency of two different hardware environments: an Nvidia A100 GPU and a Google Colab T4 GPU. Both platforms achieved similar accuracy levels on the training task. However, the A100 GPU showed significantly faster training speed, completing the task in about 15 s, while the T4 GPU took approximately 40 s.

This result underscores the A100 GPU's superior processing power, making it a suitable option for large-scale language model training when computational efficiency is crucial. On the other hand, the T4 GPU provides a more accessible and cost-effective alternative for smaller-scale projects or initial model development stages, where training time may be a less of a concern (Fig. 35.2).

The training process for our fine-tuned large language model led to a minimum loss of 0.3856 on the training set after 310 epochs. This suggests a steady improvement in the model's performance as the number of epochs increases, which aligns with the common understanding that exposing a model to more training data improves its proficiency in a given task. It's important to recognize that the ideal training loss value varies depending on the specific task and dataset used. While a lower training loss typically indicates better adaptation to the training data, it's crucial to also monitor validation loss concurrently to avoid overfitting. By monitoring both training and validation loss, we can ensure the model's capability to generalize effectively to unseen data.

Figure 35.3 provides a comparison of various machine learning models' performance on a specific task, measured by the number of accurately answered queries. Our proposed model, llama2_sharded, demonstrates superior performance, showing a notable 2.67% improvement in accuracy compared to the second-best model, zephyr3b_shard, with 477.5 correct answers compared to 465. It's important to note

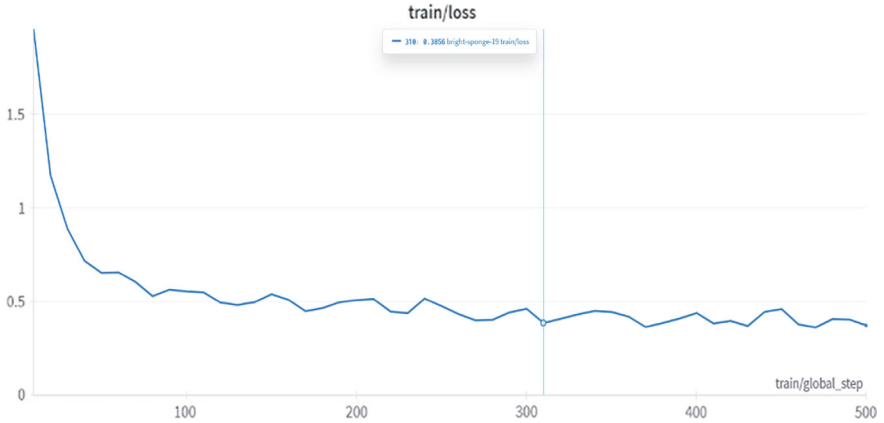

Fig. 35.2 Train loss

that the margin between llama2_sharded and the other models is even more significant, as all others fall over 19.5% below the 400 mark. These results, based on 50 questions each with a value of 10, totaling 500, have been verified by human evaluators. These findings strongly suggest the effectiveness of the llama2_sharded architecture for addressing banking queries.

Figure 35.4 presents a comparison of different machine learning models' performance on a specific task, measured by the difference in "CONTEXT_UNDERSTANDING (%)" from the llama2_sharded model, which serves as the baseline with a value of 0.00.

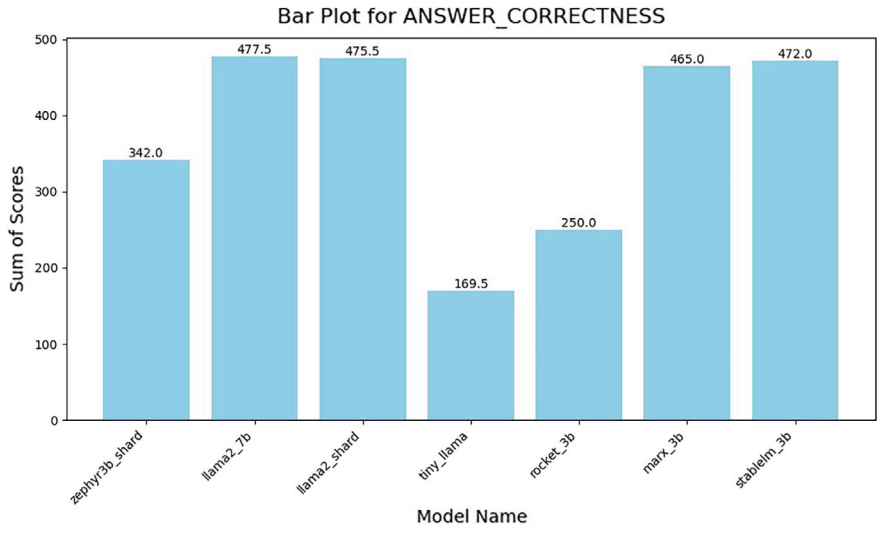

Fig. 35.3 Bar plot for answer correctness

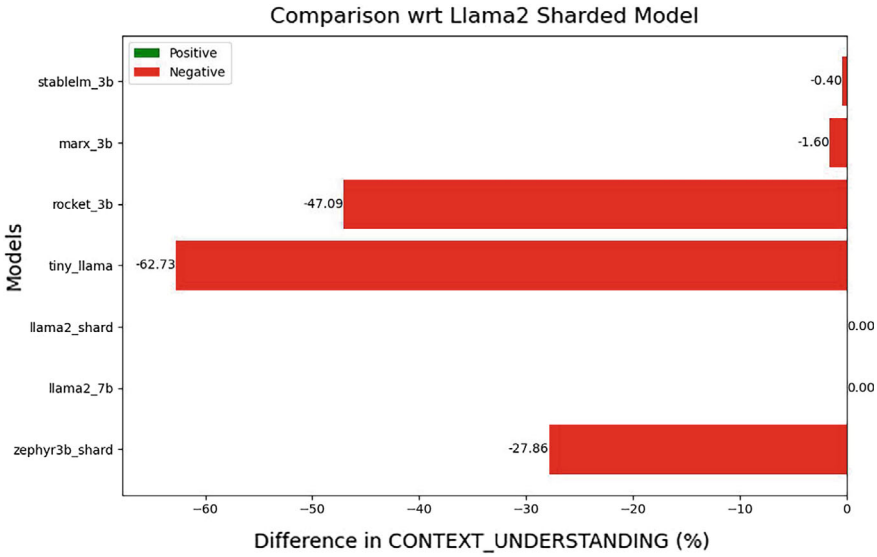

Fig. 35.4 Comparison w.r.t. Llama2_sharded model

The model labeled "stablelm_3b" shows a slight underperformance compared to the llama2_sharded model, with a CONTEXT_UNDERSTANDING score 0.4% lower. Both the "marx_3b" and "rocket_3b" models perform similarly, achieving scores around 1.6% and 47.09%, respectively, lower than the llama2_sharded model. The "tiny_llama" model exhibits a significantly lower performance, scoring 62.73% lower than llama2_sharded. The zephyr3b_shard model falls between these extremes, with a CONTEXT_UNDERSTANDING score 27.86% lower than the llama2_sharded model.

35.5 Conclusion

The training process for our fine-tuned large language model achieved a minimum loss of 0.3856 after 310 epochs, demonstrating consistent progress. This highlights the importance of sufficient exposure to training data and the necessity of monitoring both training and validation loss to prevent overfitting and ensure effective generalization to new data.

In our comparison of machine learning models for banking query tasks, fine-tuned llama2_sharded stood out for its exceptional performance. Fine-tuned llama2_sharded demonstrated a significant 2.67% improvement in accuracy compared to the second-best model, zephyr3b_shard. This resulted in fine-tuned llama2_sharded accurately answering 477.5 queries, showcasing its superiority in handling banking tasks. The notable performance gap highlights the effectiveness of fine-tuned llama2_

sharded in addressing the complexities of banking queries, suggesting its suitability for such tasks.

This superiority offers promising prospects for banking institutions in need of precise and efficient solutions for customer inquiries and transaction processing. Further exploration and optimization of fine-tuned llama2_sharded could yield even greater enhancements, meeting the evolving demands of the banking sector.

References

1. Kaplan, A., Haenlein, M.: Siri, Siri, in my hand: who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Bus. Horiz. **62**, 15–25 (2019)
2. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach. Prentice Hall, Upper Saddle River, New Jersey (2009)
3. McKinsey Consultant. What is generative AI? [Article]. Available on 7 Oct 2023. https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-generative-ai (2023)
4. Cascella, M., Montomoli, J., Bellini, V., Bignami, E.: Evaluating the feasibility of ChatGPT in healthcare: An analysis of multiple clinical and research scenarios. J. Med. Syst. **47**(1), 1–5 (2023). https://doi.org/10.1007/s10916-023-01925-4
5. Qiao, H., Liu, V., Chilton, L.: Initial images: using image prompts to improve subject representation in multimodal AI generated art. In: Proceedings of the 14th Conference on Creativity and Cognition, Venice, Italy, pp. 15–28 (2022). https://doi.org/10.1145/3527927.3532792
6. Chernyavskiy, A., Ilvovsky, D., Nakov, P.: Transformers:"the end of history" for natural language processing?. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, 13–17 Sept, Proceedings, Part III 21, pp. 677–693. Springer, Bilbao, Spain (2021)
7. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: Superglue: a stickier benchmark for general-purpose language understanding systems. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
9. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
10. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
11. Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299 (2022)
12. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
13. A. Pal, D. Karkhanis, M. Roberts, S. Dooley, A. Sundararajan, and S. Naidu, "Giraffe: Adventures in expanding context lengths in LLMs. arXiv preprint arXiv:2308.10882 (2023)
14. Chui, M., Roberts, R., Yee, L.: Generative AI is here: how tools like ChatGpt could change your business. Quantum Black AI by McKinsey, 20 Dec 2022. https://www.mckinsey.com/capabilities/quantumblack/our-insights/generative-ai-is-here-how-tools-like-chatgpt-could-change-your-business/ (2022)

15. Kasneci, E., et al.: ChatGPT for good? On opportunities and challenges of large language models for education. Learn. Individ. Differ. **103**, 1–9 (2023). https://doi.org/10.1016/j.lindif.2023.102274
16. Kung, T.H., et al.: Performance of ChatGPT on USMLE: potential for AI-assisted medical education using large language models. PLoS Digit. Health **2**(2), 1–12 (2023). https://doi.org/10.1371/journal.pdig.0000198
17. Arango, L., Singaraju, S.P., Niininen, O.: Consumer responses to AI-Generated charitable giving ads. J. Advertising, 1–18 (2023). https://doi.org/10.1080/00913367.2023.2183285
18. Lyu, K., Zhao, H., Gu, X., Yu, D., Goyal, A., Arora, S.: Keeping LLMs aligned after fine-tuning: the crucial role of prompt templates. Computer Science Department & Princeton Language and Intelligence, Princeton Univeristy. Institute for Interdisciplinary Information Sciences, Tsinghua University. arXiv:2402.18540v1 (2024)
19. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? Shanghai Key Laboratory of Intelligent Information Processing, Fudan University School of Computer Science. Fudan University, 825 Zhangheng Road, Shanghai, China
20. Touvron, H., et al.: Llama 2: Open foundation and fine-tuned chat models, 19 Jul. arXiv:2307.09288v2 (2023)
21. McKinsey Consultant. What is AI? [Article]. Available on 7 Oct 2023. https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-ai (2023)