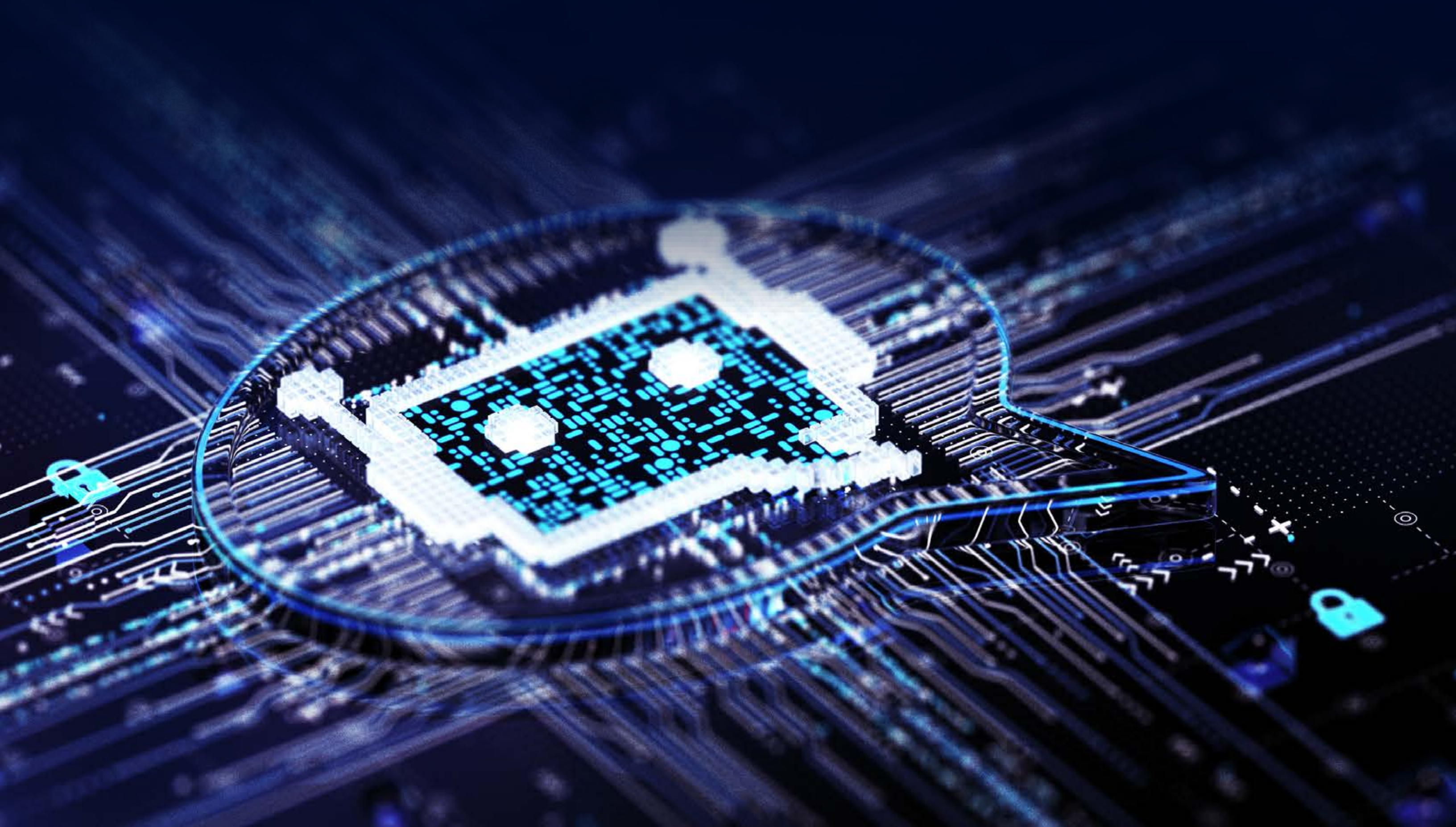
LTIMindtree

Whitepaper

Framework for Data Generation and LLM Finetuning Evaluation



Abstract

This paper presents the results of LTIMindtree's comprehensive study on fine-tuning Large Language Models (LLMs) to develop an advanced Artificial Intelligence (AI) chatbot, specifically tailored to provide information on AI policies, laws, and regulations. To showcase the efficiency of using a larger LLM for creating labeled data from raw text, AI policy data from 22 countries was used. This data preparation framework applies to various types of organizational data, including HR, finance, and medical policies. The methodology encompassed data preparation, data augmentation, and various fine-tuning techniques using platforms like Bedrock, SageMaker Jumpstart, and Amazon SageMaker. Our 2-month research highlighted the importance of smaller models, like Llama 3.2 1B, for cost-effective and efficient deployment. Quantitative and qualitative LLM evaluation metrics demonstrated the superior performance of fine-tuned models over base models using quantitative and qualitative evaluation techniques.

Introduction

In the rapidly evolving field of Artificial Intelligence (AI), applying fine-tuning techniques to Large Language Models (LLMs) is essential for achieving high performance in specialized tasks. This paper aims to study different approaches for fine-tuning large language models and develop an AI chatbot that utilizes fine-tuned LLMs to provide correct and context-specific responses across various domains. Fine-tuned models play a critical role in understanding domain-specific knowledge, which is necessary for providing accurate and relevant information. By fine-tuning LLMs on specific datasets related to a specific domain, the chatbot can better understand the nuances and complexities unique to that domain. This approach enhances the model's ability to generate responses that are contextually appropriate and informative. Additionally, the use of smaller models like Llama 3.2 1B offers advantages in terms of cost and efficiency. Despite their smaller size, these models can be fine-tuned to achieve high performance in domain-agnostic tasks. The reduced computational requirements of smaller models lead to lower inference costs, making them a cost-effective solution for deploying AI chatbots at scale. This efficiency does not compromise performance, as fine-tuning allows these models to maintain a high level of accuracy and relevance in their responses.

Challenge

Research on chatbots using LLMs, like Llama 3.2 1B, has focused on overcoming challenges in fine-tuning, particularly concerning labeled data and its preparation. The foundational transformer architecture, introduced in "Attention Is All You Need," is key to these models and the attention mechanism enhances the quality of generated responses by focusing on relevant parts of the input sequence . This helps Llama-based chatbots handle complex conversations effectively, even with limited data.

Comparative studies show that fine-tuning generally outperforms prompt engineering for specialized tasks but requires extensive labeled datasets. For instance, fine-tuning yields better performance in clinical note classification , emphasizing the need for accurate domain-specific data.

The present paper addresses this challenge by adopting a unique framework for generating data suitable for fine-tuning LLMs. It also discussed the different model evaluation frameworks based on quantitative and qualitative methods. These techniques help provide a comprehensive evaluation of fine-tuned LLMs to better gauge their capability.

The LLM finetuning techniques that were studied during the study and the observations on the pros and cons of these techniques are listed below:

Bedrock Finetuning

Bedrock-based finetuning customizes pre-trained models for specific tasks using additional training data. It can be done through User Interface (UI)-based or Software Development Kit (SDK)-based methods. The UI method is user-friendly, allowing configuration via a graphical interface, whereas the SDK method offers more control and flexibility through coding. However, it has limitations with model availability and high costs.

SageMaker JumpStart

SageMaker JumpStart fine-tunes Llama 3.2 models for specific domains using additional data . It provides a comprehensive environment through the SageMaker Studio UI or Python SDK, making the process accessible to both non-technical users and developers. The advantages include ease of use, scalability, and customization, although it lacks extensive control over data augmentation and monitoring.

Customized Training

This method allows the training of Machine Learning (ML) models using open-source models from Hugging Face. Also, it allows for customization of the data loading, transformation/augmentation, and training pipelines. It is to be noted that this approach can be implemented across different cloud platforms, such as Amazon Web Services (AWS), Azure, Google Cloud Platform (GCP), etc., as per the needs of the organization. It supports a wide range of pre-trained models for various Natural Language Processing (NLP) tasks, enabling users to leverage advanced research in their applications.

Solution Approach

1. Data Preparation

To showcase the efficiency of using a larger LLM for creating labeled data from raw text, AI policy data from 22 countries was used . This data preparation framework applies to various types of organizational data, including HR, finance, and medical policies. Web scraping with Beautiful Soup gathered AI regulations from master policy documents across nations. The content was converted into PDFs, then turned into question-answer pairs using the Claude Sonnet 3 Model on AWS Bedrock, resulting in JSON Lines (JSONL) formatted data. Figure 1 illustrates the data extraction process.

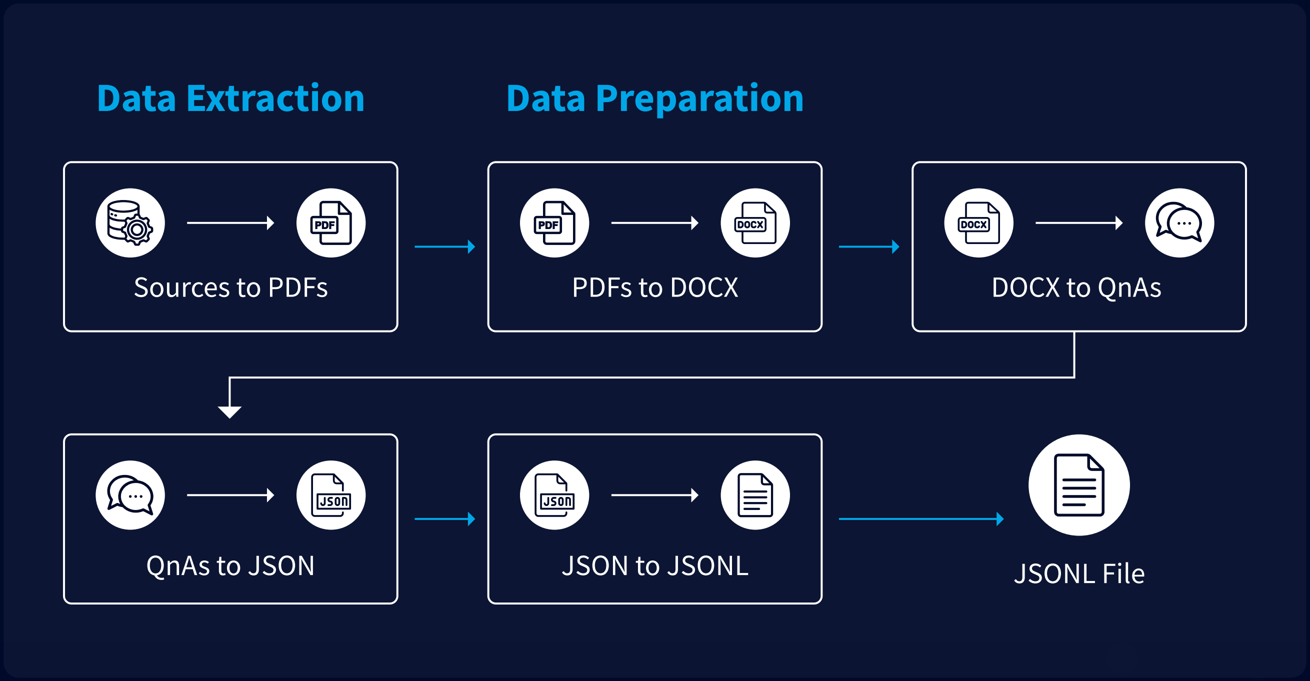


Figure 1. *Flowchart showing Data Preparation and EDA of the Text Data*

2. Customized Training Using Sagemaker

Given the flexibility and cost advantages that customization offers, this method was adopted to fine-tune the Llama 3.2 1B model on AWS SageMaker. Using a SageMaker notebook over an EC2 instance offers flexible resource allocation. Code development and experimentation can be done on a smaller instance, while a more powerful instance with higher GPU capacity can be attached for model training. This dynamic scaling ensures resource efficiency and cost reduction.

2.1. Data Augmentation

Data augmentation is a crucial step in enhancing the robustness and diversity of the AI chatbot's training data. This section outlines the approach used for various data augmentation techniques to demonstrate the flexibility of using SageMaker for training an LLM model.

Several data augmentation techniques are implemented to enhance the training data:

- **Synonym Replacement:** Replacing words in the text with their synonyms helps generate variations of the text while preserving the original meaning.
- **Back Translation:** Back translation involves translating the text to another language and then back to the original language. This technique introduces variations in sentence structure and wording, enhancing the diversity of the training data.
- **Random Character Deletion:** Randomly deleting characters from words in the text, simulates typographical errors. It helps create variations that can improve the model's robustness to such errors.
- **Random Character Swap:** This method swaps neighboring characters in words, simulating typographical errors. It introduces variations that can help the model handle input errors.

The overall approach involves randomly selecting one of these augmentation techniques and applying it to the input text. This random selection ensures that the training data is diverse and robust, covering various types of augmentations. The dataset is then loaded, shuffled, and split into training and testing sets. Each set is further divided into multiple parts to facilitate the augmentation process using lesser memory. Importantly, data augmentation is applied only to the training data, not to the validation data (referred to as test data in this context). The augmented training data is combined with the original training data to create a comprehensive dataset. Both the augmented and original datasets are then saved into JSONL files for further use in model training and fine-tuning. This structured approach ensures that the training data for the AI chatbot is enriched with diverse and robust variations, ultimately enhancing the model's performance and generalization capabilities.

2.2. Model Training

The AI chatbot training involved several key steps, including:

- Loading datasets from JSON files and initializing the tokenizer and base model
- Defining training parameters, including learning rate, batch sizes, epochs, weight decay, and gradient accumulation
- Shuffling and tokenizing the dataset and creating samplers and data loaders
- Initializing the optimizer and learning rate scheduler
- Running the training loop for the specified epochs and processed batches
- Computing outputs and losses
- Updating optimizer and scheduler with Low-Rank Adaptation (LoRA) and Distributed Data Parallel (DDP) for distributed training

After each epoch, the model is evaluated based on validation data and saved if the evaluation loss improves, resetting the early stopping counter. If no improvement was observed, early stopping was triggered. Additionally, the model, tokenizer, and configuration files were saved. This method ensures effective fine-tuning using distributed training and optimization to enhance performance.

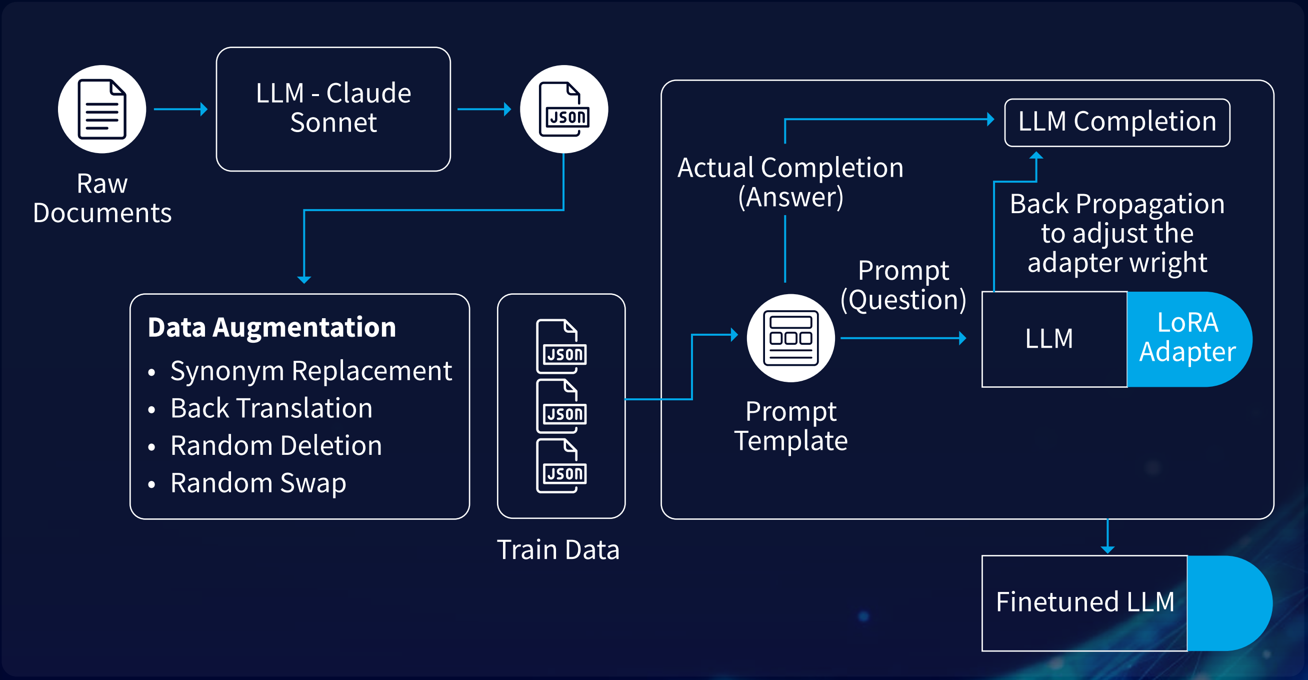


Figure 2. *LLM Finetuning Framework*

2.3. Model Evaluation

The fine-tuned model was evaluated on the test data using both quantitative and qualitative methods. To better understand the finetuned model's response, the base model's response was also generated to run a comparative analysis of the metrics on the test data (unseen by the model). This data was extracted from the frequently asked questions sections of various policy websites and documents through web scraping and was not used for fine-tuning the model. Below is the list of LLM evaluation metrics and methods that were used to evaluate the performance of fine-tuned LLM.

2.3.1. Quantitative Evaluation

1. **Relevance Score b/w Tokenized Vectors:** The relevance score measures the semantic similarity between two pieces of text. In this study, this score was calculated based on the question and the model's response.

2. Cosine Similarity: It ranges from -1 to 1 and was calculated between the actual response and the model's response:

- 1 indicates identical direction which means, the responses form similar meaning or context
- 0 indicates no similarity (orthogonal) which means, the responses that are not related to each other
- -1 indicates the opposite direction which means, the responses are opposite in meaning or context

3. **BERT Score Evaluation:** It measures text similarity using contextual embeddings from BERT. Unlike traditional evaluation metrics like BLEU or ROUGE, it compares each token in the model response text with each token in the actual response text, calculating precision, recall, and F1 score (harmonic mean of precision & recall) based on these similarities.

2.3.2. Qualitative Evaluation

Responses from both the fine-tuned and base models were compared using the Claude Sonnet model. This provided a detailed analysis and insights into the response based on the question provided and the actual response that was web-scrapped when compared to the responses by the base and fine-tuned LLM.

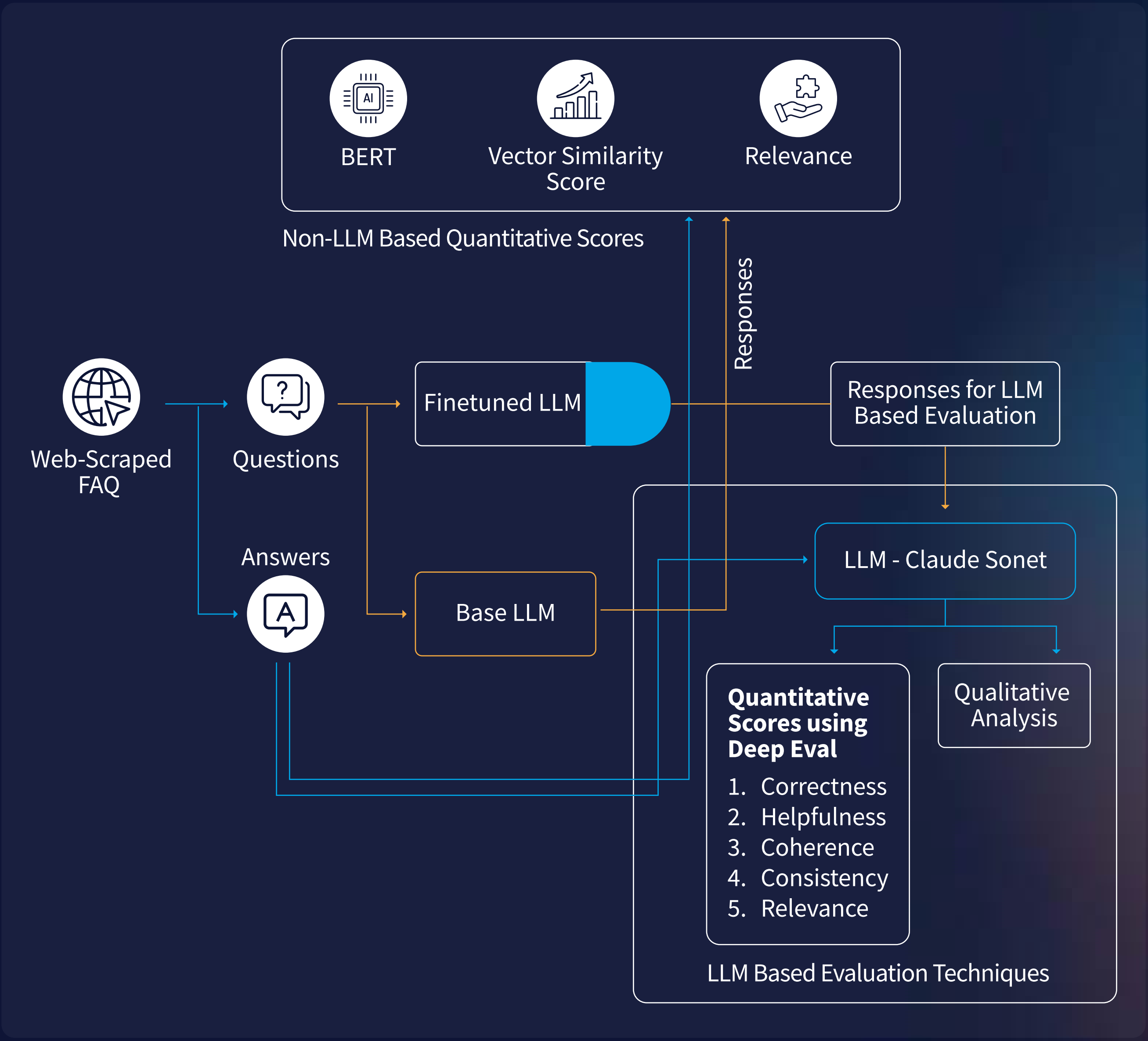


Figure 3. *LLM Evaluation Framework*

Results

The non-LLM based qualitative analysis (using BERT, Vector Similarity and Relevance Scores) of the responses from both finetuned and base Llama models were compared as shown in table 1.

Stats	Finetuned			Base		
	BERT	Cosine Similarity	Relevance	BERT	Cosine Similarity	Relevance
mean	0.69	0.69	0.73	0.68	0.68	0.73
std	0.10	0.10	0.09	0.12	0.12	0.09
min	0.49	0.49	0.59	0.40	0.40	0.59
25%	0.63	0.63	0.67	0.60	0.60	0.69
50%	0.72	0.72	0.71	0.71	0.71	0.72
75%	0.77	0.77	0.78	0.77	0.77	0.76
max	0.84	0.84	0.93	0.83	0.83	0.92

Table 1. *Comparison of qualitative metrics for fined-tuned and base Llama models*

It can be inferred that both models performed equally on the unseen test data. However, to better understand the performance nuances, DeepEval-based LLM evaluation metrics (except Fluency), as defined below, were also calculated for the same test data.

Correctness

- Evaluates whether the actual output is factually correct based on the expected output
- Example criteria: "Determine whether the actual output is factually correct based on the expected output"

Helpfulness

- Measures how useful the response is to the user
- Example criteria: "Assess whether the response provides useful and relevant information to the user's query"

Relevance

- Checks if the response is relevant to the input query
- Example criteria: "Evaluate whether the response directly addresses the input query"

Fluency

- Evaluates the grammatical correctness and readability of the response
- Example criteria: "Determine whether the response is grammatically correct and easy to read"

Coherence

- Measures the logical flow and consistency of the response
- Example criteria: "Assess whether the response is logically structured and consistent"

Consistency

- Checks for factual alignment between the response and the source document
- Example criteria: "Evaluate whether the response contains only statements that are entailed by the source document"

Tables 2 and 3 show that finetuning large language models significantly improves Llama's smaller models, even with public data. This demonstrates the effectiveness of this framework for task specificity and highlights the importance of small LLMs like Llama 3.2 1B in developing cost-effective solutions.

Stats	Correctness (Deep Eval)	Helpfulness (Deep Eval)	Relevance (Deep Eval)	Coherence (Deep Eval)	Consistency (Deep Eval)
mean	0.50	0.66	0.57	0.79	0.59
std	0.28	0.16	0.25	0.14	0.23
min	0.00	0.30	0.00	0.30	0.00
25%	0.30	0.50	0.50	0.80	0.50
50%	0.50	0.70	0.50	0.80	0.50
75%	0.80	0.80	0.70	0.80	0.80
max	0.80	0.80	1.00	1.00	1.00

Table 2. *Finetuned Llama metrics as measured using the DeepEval method*

Stats	Correctness (Deep Eval)	Helpfulness (Deep Eval)	Relevance (Deep Eval)	Coherence (Deep Eval)	Consistency (Deep Eval)
mean	0.47	0.62	0.58	0.75	0.54
std	0.28	0.17	0.27	0.21	0.22
min	0.00	0.20	0.00	0.00	0.00
25%	0.25	0.50	0.50	0.80	0.50
50%	0.50	0.60	0.50	0.80	0.50
75%	0.80	0.80	0.80	0.80	0.75
max	0.80	0.80	1.00	1.00	0.80

Table 3. *Base Llama metrics as measured using the DeepEval method*

The qualitative analysis comparing the base and fine-tuned models' responses with actual website data shows:

- **Sentiment analysis:** Finetuned model responses are more structured, meaningful, and accurate than the base model, aligning closely with actual responses.
- **General questions:** Finetuned model responses are concise and relevant, while base model responses are generic and lack detail.
- **Summary questions:** Finetuned model provides detailed, accurate summaries, whereas the base model lacks depth.
- **Legal and regulatory questions:** Finetuned model responses are accurate and aligned with legal provisions; base model responses often contain inaccuracies.
- **Overall performance**: The finetuned model consistently outperforms the base model in structure, relevance, and reliability, capturing key points more accurately.

Key Highlights

- **Superior Performance:** The fine-tuned Llama model outperforms the base model in various question types, providing more structured, meaningful, and relevant responses.
- **Accurate Nuance Capture:** The fine-tuned model demonstrates an improved ability to capture key points and nuances.
- **Enhanced Context-Specific Guidance:** Properly fine-tuning large language models with relevant data significantly improves performance and reliability in providing context-specific guidance.
- **Contextually Appropriate Responses:** Fine-tuned models provide more contextually appropriate and informative responses, which is crucial for applications like AI chatbots handling complex queries.
- **Improved Efficiency and Effectiveness:** Fine-tuning LLMs to specific domains improves the accuracy and relevance of AI applications, leading to better customer engagement and satisfaction.
- **Cost-Effective Solutions:** Using smaller models like Llama 3.2-1B ensures cost-effective deployment without compromising performance, making it feasible for businesses to scale their AI solutions.
- **Reduced Computational Costs:** Fine-tuning reduces computational costs and accelerates the development and deployment of AI applications, providing a competitive edge in the market.
- **Cost-Effective Updates:** The fine-tuning process, facilitated by AWS SageMaker, can be repeated with new data at minimal cost and time, ensuring the model remains up-to-date.
- **Periodic Updates:** The ability to periodically update fine-tuned models with new data ensures that AI solutions remain up-to-date and relevant, enhancing their long-term value and reliability.
- **Data Security:** Fine-tuning on proprietary data addresses data security concerns while enhancing chatbot effectiveness.
- **Importance of Frameworks:** The success underscores the importance of robust data generation, fine-tuning, and evaluation frameworks in achieving superior model performance.

The success of this study indicates that fine-tuning a pre-trained LLM with relevant data can significantly enhance its performance and reliability in providing context-specific guidance. The fine-tuned model aligns better with actual responses, making it a more powerful tool for navigating complex regulatory landscapes.

Conclusion

The present study has shown that the fine-tuned Llama model significantly outperforms the base Llama model across various question types. The fine-tuned model provides responses with enhanced structure, meaningfulness, and relevance, demonstrating its ability to capture key points and nuances more accurately. Notably, this improvement is evident even though most of the documents used in this study were publicly available and already part of Llama 3.2 1-B's training data. This approach highlights the potential of fine-tuning models on domain-specific, proprietary data to create more effective chatbots while addressing data security concerns. Additionally, the fine-tuning process, facilitated by AWS SageMaker, can be periodically repeated with new data at minimal cost and time, thanks to the small size of the LLM and the flexibility in deploying such models cost-effectively.

References

1. [i]Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems, 2017.
2. [ii] "The Challenges of Evaluating LLM Applications: An Analysis of Automated, Human, and LLM-Based Approaches."
3. iii Navule, P. K. R., & Ruan, J. T. (2024, November 11). Fine-tune Meta Llama 3.2 text generation models for generative AI inference using Amazon SageMaker JumpStart. AWS Machine Learning Blog. Retrieved from https://aws.amazon.com/blogs/machine-learning/fine-tune-meta-llama-3-2-text-generation-models-for-generative-ai-inference-using-amazon-sagemaker-jumpstart/
4. iv IAPP. 2024. Global AI Law and Policy Tracker. IAPP. Available at: https://iapp.org/media/pdf/resource_center/global_ai_law_policy_tracker.pdf

Whitepaper Page Synopsis according to the new template

The present study outlines a comprehensive approach to fine-tuning Large Language Models (LLMs) for developing an AI chatbot tailored to provide information on AI policies, laws, and regulations. It emphasizes the importance of data preparation, data augmentation and fine-tuning techniques using platforms like Bedrock, SageMaker Jumpstart, and Amazon SageMaker. The study highlights the cost-effectiveness and efficiency of smaller models like Llama 3.2-1B, demonstrating their superior performance through quantitative and qualitative evaluations.

Key challenges addressed include acquiring high-quality labeled data and the benefits of customized training methods. The solution approach involves data augmentation, model training, and evaluation, showcasing significant improvements in the fine-tuned model's performance and reliability. The paper concludes that fine-tuning LLMs on domain-specific data enhances their ability to provide context-specific guidance, with the process being cost-effective and easily repeatable using AWS SageMaker.

About the author

Bharath Adapa

Associate Principal, Enterprise AI,
LTIMindtree

With 13 years of experience in AI and ML, Bharath Adapa specializes in developing advanced AI-driven solutions across industries like manufacturing, marketing, risk management, and bio-pharma. Proficient in Python, TensorFlow, PyTorch, and AWS, he has successfully led projects that enhance data retrieval, predictive analytics, and intelligent automation. His expertise spans deploying graph databases, fine-tuning LLMs, and building high-accuracy ML models for complex challenges.

Holding a Master's in ML & AI from Liverpool John Moore's University and a B.Tech in Engineering Physics from IIT Delhi, Bharath is deeply passionate about AI research and innovation.

Bharath Adapa
Associate Principal - Data Sciences, Enterprise AI
adapa.bharath@ltimindtree.co

Sateesh Gottumukkala
Senior Director - Data Sciences, Enterprise AI
sateesh.gottumukkala@ltimindtree.com

Yash Kumar Kalyan
Consultant - System Management, Enterprise AI
Email: yash.kalyan@ltimindtree.com

Neha Saxena
Consultant - System Management, Enterprise AI
neha.saxena2@ltimindtree.com