Building an E-Commerce FAQ Chatbot Using Parameter Efficient Fine Tuning with LoRA Technique

Bal Narendra Sapa
University of New Haven
New Haven, CT 06511
bsapa1@unh.newhaven.edu

Ajay Kumar Jagu
University of New Haven
New Haven, CT 06511
ajagu1@unh.newhaven.edu

Abstract

This project aims to develop a FAQ chatbot for an E-Commerce site by leveraging Large Language Models (LLMs). We adopt a fine-tuning approach using a small-size model, Falcon-7B, and employ Parameter Efficient Fine Tuning (PEFT) with LoRA Technique to enhance the model's performance. The chatbot is trained on a custom dataset extracted from a Kaggle dataset, addressing common user queries. The fine-tuned model is made available on Hugging Face, accompanied by a comprehensive Jupyter notebook providing code and explanations. The potential of hosting a Streamlit app is explored, considering resource limitations.

1 Introduction

In the fast-paced world of e-commerce, numerous websites cater to a massive influx of customers seeking to purchase a variety of items. Naturally, users often have questions related to the buying process, account management, refunds, and other common queries. To address these concerns, customers typically turn to the customer care section, where dedicated agents work tirelessly to provide solutions. However, the nature of these inquiries tends to be repetitive, leading to a considerable need for customer support staff. Large companies, in particular, may find themselves employing thousands of workers solely to handle these routine questions, incurring significant operational costs that can run into millions of dollars.

Enter the potential solution: leveraging advanced language models that are fine-tuned on the specific questions and answers prevalent in the e-commerce domain. By employing such a model, companies can streamline their customer support processes, allowing the automated system to handle repetitive and general queries. This not only enhances efficiency but also presents a cost-effective alternative, potentially saving companies substantial amounts of money previously spent on maintaining extensive customer support teams. As these language models can swiftly and accurately respond to common inquiries, customer care agents can then focus on more complex and nuanced issues, ultimately providing a more efficient and cost-conscious approach to managing customer support in the dynamic realm of e-commerce.

2 Objective

2.1 Efficient Customer Support

Leveraging a Large Language Model allows companies to streamline customer support by automating responses to frequently asked questions. This not only enhances the efficiency of addressing common queries but also ensures that customers receive prompt and accurate information, leading to improved overall satisfaction.

.

2.2 Cost Savings through Automation

Implementing a Large Language Model in customer support operations helps companies cut costs significantly. By automating responses to routine inquiries, organizations can reduce the workload on human support agents, enabling them to focus on more complex and personalized customer interactions. This automation translates into substantial savings for companies heavily reliant on customer support services.

2.3 Enhanced Resource Allocation

With the integration of a Large Language Model in customer support, companies can allocate human resources more strategically. By automating responses to FAQs, human agents can concentrate on tasks that require creativity, empathy, and problem-solving skills, ultimately improving the overall quality of customer service. This optimized resource allocation contributes to a more effective and economical customer support system.

3 Related Work

3.1 Pre-Trained Models

3.1.1 Open AI's GPT models

Fine-tuning OpenAI's GPT models has become a prevalent practice in natural language processing tasks due to the remarkable capabilities of these language models. GPT-3, GPT-3.5, and GPT-4, being powerful language models (LLMs), offer a versatile foundation for various applications.

3.1.2 LLAMA models by META

In recent developments, META has introduced the LLAMA models, which are on par with OpenAI's GPT models in terms of capability. Notably, a significant distinction lies in the accessibility and openness of LLAMA models. Unlike the closed nature of OpenAI's models, LLAMA models are open source, allowing researchers and developers to download and fine-tune them for specific use-cases.

Fine-tuning of LLAMA models involves training on domain-specific datasets, similar to the process with GPT models. This adaptability makes LLAMA a compelling alternative for those who seek not only powerful language models but also the freedom to tailor these models to their specific needs.

3.2 ChatBots

3.2.1 IBM Watson Assistant

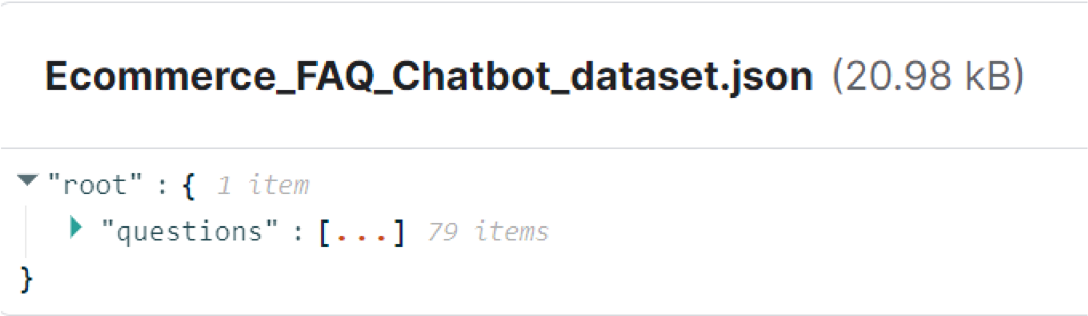
IBM Watson Assistant provides a robust platform to create customized live chatbots tailored for diverse applications. These chatbots, often referred to as assistants, can seamlessly integrate into any device, application, or channel, offering a versatile solution for engaging with users across different touchpoints.

3.2.2 Ada Healthcare Chatbot

The Ada Health chatbot is a free app that uses AI to provide personalized medical advice and symptom evaluations. The chatbot asks users questions about their health and symptoms, and then assesses the answers against a medical dictionary. The result is a personalized report that indicates what could be wrong and what the user could do next.

3.3 RAG vs Fine-Tuned Model

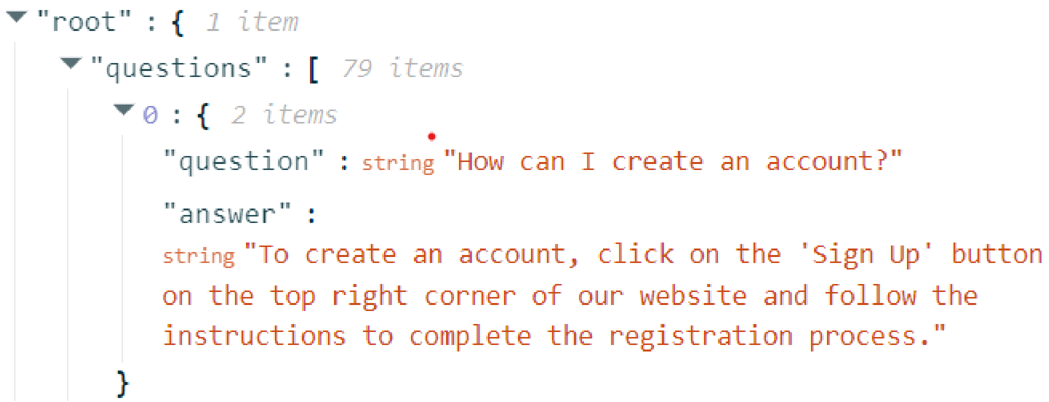
Introducing a chatbot for frequently asked questions (FAQs) can be a money-saver for companies because it reduces the need for a lot of customer support. This is especially beneficial for companies that rely heavily on customer support.



```
Ecommerce_FAQ_Chatbot_dataset.json (20.98 kB)

▼ "root" : { 1 item
    ▶ "questions" : [...] 79 items
}
```

Figure 1: Dataset in Kaggle.



```
▼ "root" : { 1 item
    ▼ "questions" : [ 79 items
        ▼ 0 : { 2 items
            "question" : string "How can I create an account?"
            "answer" :
            string "To create an account, click on the 'Sign Up' button
            on the top right corner of our website and follow the
            instructions to complete the registration process."
        }
```

Figure 2: Sample 1.

Some advanced chatbots, like those using Retrieval Augmented Generation (RAG), don't actually learn from private company data. Instead, they use pre-made FAQs and create special codes (embeddings) for them. When you ask a question, these codes and your question go to a smart language model (LLM), which then generates an answer based on the codes.

The issue here is that the LLM forgets everything about the FAQ after answering. For big companies, this means extra work because they have to keep sending codes to the model for each question. Plus, storing and managing these codes can be a hassle.

On the other hand, a fine-tuned model, which is trained on the company's private data, doesn't have this problem. It can answer questions on its own without needing old data. If the company fine-tunes the model every month with their own data, it stays updated without the need for creating and managing those special codes. This approach is not just convenient but also cheaper than dealing with all the coding work.

4 Dataset

The dataset used in this project is sourced from Kaggle and is open-source, meaning there are no restrictions on its use. It comprises 79 samples, each consisting of a question and its corresponding answer.

To access the dataset, simply click on the provided link in this section, which will direct you to the dataset on Kaggle.

The dataset splilt is shown in Table [1]

`https://www.kaggle.com/datasets/saadmakhdoom/ecommerce-faq-chatbot-dataset?select=Ecommerce_FAQ_Chatbot_dataset.json`

Table 1: Dataset Split

Dataset	Number of Question and Answers
Train	67 (85 %)
Test	12 (15 %)

5 Methodology

This part talks about what we need to make the model better and how we make it work better. It explains the tools, computer power, and data we use for this process, giving a clear picture of how we improve the FALCON-7B model. It also explores the methods we use to fine-tune the model, showing how we strategically improve its abilities and customize it to meet specific needs.

5.1 Pre-Trained Model

We have chosen the FALCON-7B model as our foundational model, boasting an impressive size of approximately 15 gigabytes.

5.2 Resources Needed

In order to train the model, we will be utilizing Google Colab, which provides a complimentary T4 GPU boasting 15 gigabytes of VRAM (Video Random Access Memory). Alternatively, Kaggle Notebook presents another viable option, offering two T4 GPUs, each equipped with 14 gigabytes of VRAM, resulting in a combined VRAM capacity of approximately 29 gigabytes. However, it's important to note that Kaggle Notebook has resource constraints and may have limitations on usage. These selected platforms serve as the essential resources for our model training endeavors.

5.3 Techniques

In recent advancements within natural language processing (NLP) and beyond, large language models (LLMs) based on the transformer architecture, such as GPT, T5, and BERT, have emerged as state-of-the-art solutions across various tasks. However, the increasing size of these models presents challenges for fine-tuning on consumer hardware and incurs substantial computational and storage costs. To address these issues, Parameter-Efficient Fine-Tuning (PEFT) approaches have been introduced.

PEFT selectively fine-tunes a small subset of model parameters while keeping the majority frozen, significantly reducing both computational and storage requirements. This method not only mitigates the challenges associated with the impracticality of full fine-tuning on consumer-grade hardware but also overcomes issues like catastrophic forgetting, observed during conventional fine-tuning of LLMs. PEFT approaches demonstrate superiority in low-data scenarios, exhibit improved generalization in out-of-domain scenarios, and offer enhanced portability by enabling the use of a single LLM for multiple tasks through the addition of small, task-specific weights.

5.4 Libraries

The fine-tuning process involves the utilization of key libraries, namely Transformers and Peft by Hugging Face. These libraries play a crucial role in optimizing and adapting the model for specific tasks. Additionally, the Datasets library is employed to seamlessly convert the raw dataset into a format compatible with Hugging Face. The torch library is essential for handling various aspects of the model training, while Bitsandbytes is also utilized to enhance specific functionalities in the fine-tuning pipeline. Together, these libraries form an integral part of the toolkit used to refine and improve the performance of the FALCON-7B model.

Table 2: LoRA Hyperparameters

Hyperparameter	value
r	16
lora alpha	32
lora dropout	0.05

Table 3: Trainer Hyperparameters

Hyperparameter	value
epochs	5
learing rate	2e-4

5.5 Converting into Hugginface compatible dataset

We have a raw dataset that we want to use with the Hugging Face library, specifically with functions commonly used in the transformers library. To make this possible, we need to convert the raw dataset into a format that is compatible with Hugging Face.

Once the conversion is done, the dataset object will have functions that seamlessly work with the transformers library. This conversion is necessary to ensure that our dataset can be easily integrated and utilized within the Hugging Face ecosystem.

5.6 Dataset Preprocessing

The dataset obtained from Kaggle requires preprocessing to align with our specific requirements. Each sample in the provided dataset undergoes a transformation through tokenization using the designated tokenizer tailored for the FALCON-7B model. This preprocessing step ensures that the data is appropriately formatted and ready for effective utilization in the fine-tuning process, aligning with the model's architecture and specifications.

5.7 Configuring LoRA Adapters

The hyperparameters employed for LoRa Adapters are illustrated in Table [2].

5.8 Configuring Trainer

The hyperparameter employed for Trainer provided by huggingface are illustrated in Table [3]

6 Result

The model was trained as per the given instructions. We used the Bleu score to measure and assess how well the model performs. The Bleu score helps us understand how closely the model's output matches the expected or desired results.

6.1 Loss

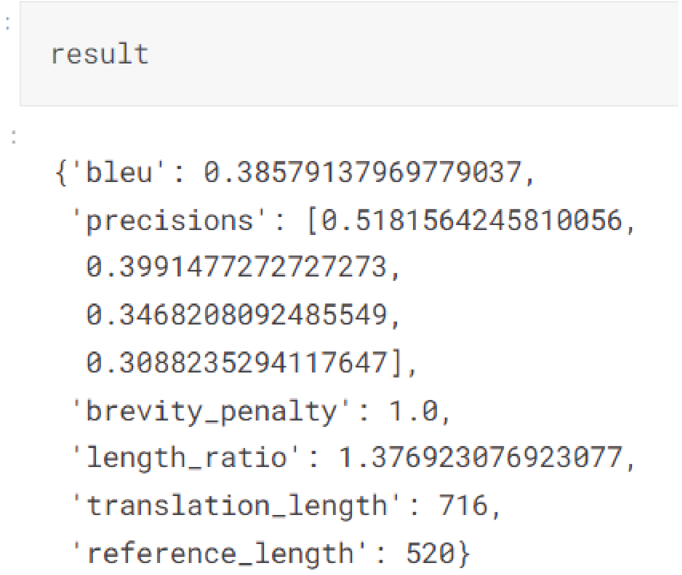
The loss across epochs for both the validation and training datasets is systematically decreasing, indicating a positive trend in model training. Detailed information on the loss values at different epochs is presented in the Table [4]

6.2 Bleu Score

The Bleu score is calculated for test dataset. It is shown in image [6.2]

Table 4: Loss

Epoch	Training loss	validation loss
1	1.1965	1.3483
2	0.7485	0.947881
3	0.562500	0.858913
4	0.4886	0.83501
5	0.6888	0.823139



```
:   result

:   {'bleu': 0.38579137969779037,
     'precisions': [0.5181564245810056,
      0.3991477272727273,
      0.3468208092485549,
      0.3088235294117647],
     'brevity_penalty': 1.0,
     'length_ratio': 1.376923076923077,
     'translation_length': 716,
     'reference_length': 520}
```

Figure 3: Bleu Score

7 Conclusion

In this project, we embarked on the development of an E-Commerce FAQ chatbot utilizing the FALCON-7B model. Leveraging the power of Large Language Models, we employed a fine-tuning approach, specifically utilizing Parameter Efficient Fine Tuning (PEFT) with LoRA Technique.

Our objective was to enhance customer support efficiency, achieve cost savings through automation, and optimize resource allocation within an e-commerce setting.

Our dataset, extracted from Kaggle, provided a diverse set of questions and answers, enabling the training of a robust model capable of handling various user queries. Through careful preprocessing and the adoption of key libraries such as Transformers and Peft by Hugging Face, we configured and fine-tuned the FALCON-7B model to align with the specific requirements of our task.

The training process demonstrated promising results, with decreasing loss values across epochs, indicating the model's ability to learn from the data. The evaluation using Bleu scores on the test dataset showcased the model's proficiency in generating responses that align with expected results.

While the achieved results are promising, it's essential to acknowledge the limitations of our study. The dataset size, while diverse, remains relatively small, and further experiments with larger datasets could potentially yield even better performance. Additionally, the nature of user queries in real-world scenarios can be dynamic and may require continuous refinement of the model over time.

In conclusion, our work contributes to the broader goal of enhancing customer support in e-commerce through the integration of advanced language models. The achieved results showcase the potential of fine-tuned models in automating responses to frequently asked questions, leading to cost savings and improved resource allocation. As we move forward, it is imperative to explore continuous model refinement, address scalability concerns, and adapt to evolving user needs in the dynamic landscape of e-commerce.

8 Acknowledgements

We would like to express our sincere gratitude to everyone who contributed to the successful completion of this project.

First and foremost, we extend our thanks to the developers and contributors of the FALCON-7B model and the associated libraries, particularly the Hugging Face community. Their open-source contributions provided the foundation for our project, enabling us to work with cutting-edge language models and fine-tuning techniques.

We are indebted to Kaggle for hosting the open dataset that formed the basis of our training data. The availability of diverse and relevant samples greatly facilitated the development of a robust E-Commerce FAQ chatbot.

Our appreciation goes to the faculty and staff at the University of New Haven, who provided an intellectually stimulating environment for academic exploration and research. Special thanks to Vahid Behzadan, our course instructor, for their valuable insights, guidance, and continuous support throughout the project.

9 Deployment

We've developed a Streamlit application that you can use on your own computer. However, for the application to run at its best, it requires a graphics processing unit (GPU) with at least 16 gigabytes of video RAM (vRAM). it is show in image [9]

To make sure the application utilizes the GPU properly, there are certain sections in the Streamlit file that are commented out. Uncommenting these sections will activate the GPU support, allowing the application to take full advantage of the available graphics power.

10 Code and Resources

The code for this project, along with the fine-tuned model, is publicly available for reference and use. You can find the code repository on GitHub at the following link

`https://github.com/balnarendrasapa/faq-llm`

Additionally, the fine-tuned model is accessible on the Hugging Face Model Hub:

follow this link:

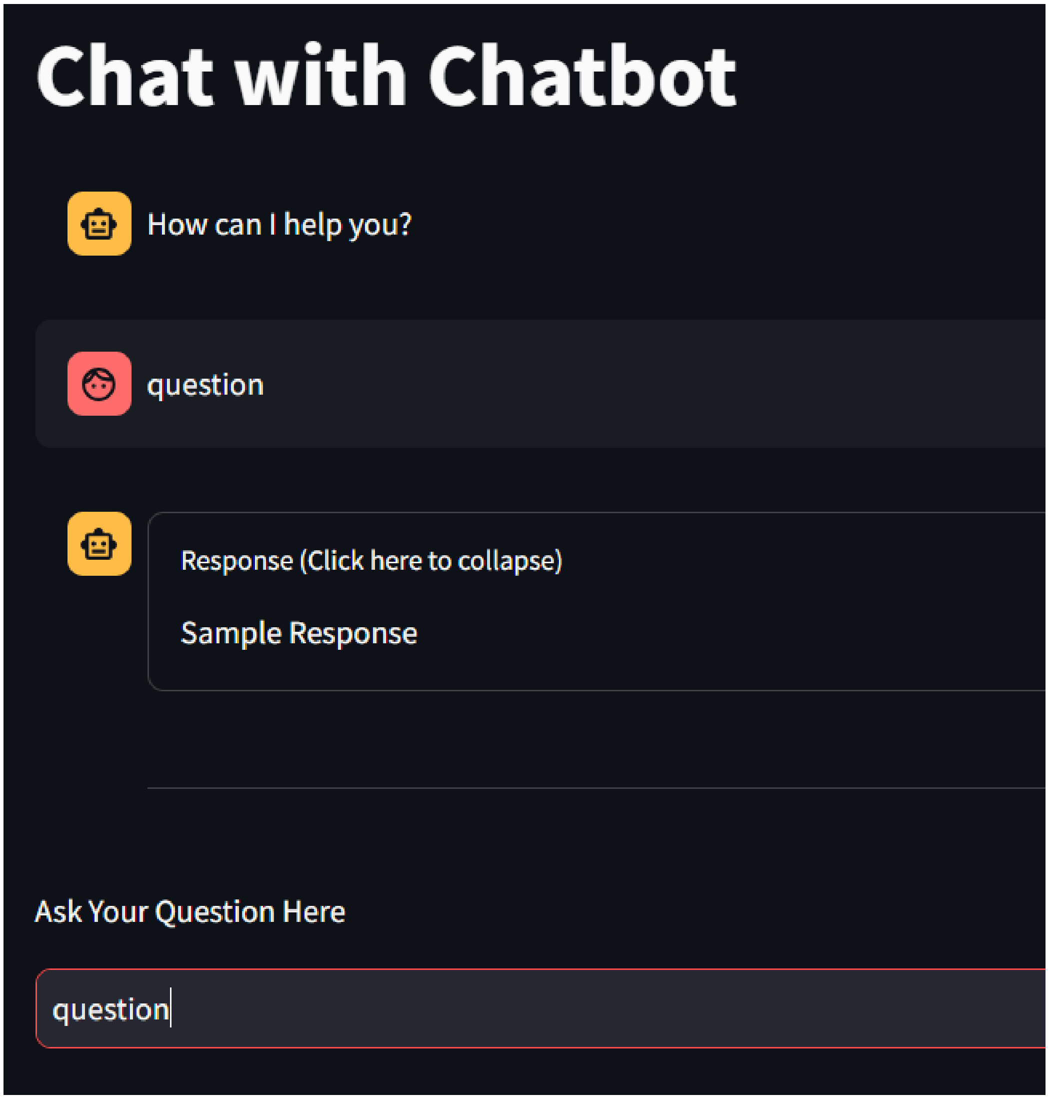
`https://huggingface.co/bnsapa/faq-llm`

For a detailed walkthrough and interactive exploration, we have provided a Kaggle notebook that guides users through the code implementation and offers a hands-on experience.

The Kaggle notebook can be accessed through the following link:

`https://www.kaggle.com/code/balnarendrasapa/`
`fine-tuning-falcon-7b-with-faq-e-com-dataset`

Feel free to explore the codebase, experiment with the model, and contribute to further improvements. We encourage collaboration and welcome feedback from the community.



Chat with Chatbot

How can I help you?

question

Response (Click here to collapse)

Sample Response

Ask Your Question Here

question

Figure 4: Stremlit app

References

[1] Tom B. Brown et al (2020) Language Models are Few-Shot Learners. arxiv id:2005.14165

[2] LLAMA 2 Model (2023) Llama 2: Open Foundation and Fine-Tuned Chat Models. arxiv id:2307.09288

[3] Haokun Liu et al (2022) Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. arxiv id:2205.05638

[4] Edward J Hu et al (2021) LoRA: Low-Rank Adaptation of Large Language Models. arxiv id:2106.09685

[5] James Manyika (2023) An overview of Bard: an early experiment with generative AI. link - https://ai.google/static/documents/google-about-bard.pdf

[6] IBM Watson Assistant. link - https://www.ibm.com/products/watsonx-assistant

[7] Sourab Magrulkar, Sayak Paul (2023). link - https://huggingface.co/blog/peft