Large Language Models in Consumer Electronic Retail Industry: An AI Product Advisor

Loo Seng Xian[1]; Lim Tong Ming[2*]
Tunku Abdul Rahman University of Management and Technology
Kuala Lumpur, Malaysia

Corresponding Author: Lim Tong Ming[2*]

Abstract:- This paper explores the development of an AI product advisor utilizing large language models (LLMs). Firstly, we discuss the needs and the current problems faced by industry. Subsequently, we reviewed past works by various scholars regarding AI Assistant in Retail and Other Industries, and LLM Models and techniques used in generative AI chatbot for sales and service activity related works. Next, we assessed the performance of various models including Llama2B, Falcon-7B, and Mistral-7B, in conjunction with advanced response generation techniques such as Retrieval Augmented Generation (RAG), fine-tuning through QLora and LLM chaining. Our experimental findings reveal that the combination of Mistral-7B with the RAG and LLM chaining technique enhances both efficiency and the quality of model responses. Among the models evaluated, Mistral-7B consistently delivered satisfactory outcomes. We deployed a prototype system using Streamlit, creating a chatbot-like interface that allows users to interact with the AI advisor. This prototype could potentially increase the productivity of frontliners in the retail space and provide benefits for the industry.

Keywords:- *Mistral-7B, Llama-2, Falcon-7B Generative AI, Generation AI, Language Model-based AI, Retrieval Augmented Generation (RAG), QLora, LangChain*

I. INTRODUCTION

A. Background of the Consumer Electronics Retail Industry

The consumer electronic industry has seen a significant transformation in consumer shopping experience at the retail outlets; evolving from mere sales points to sophisticated experience centers (Reinart et al., 2019). These modern retail environments are designed to allow customers to interact with and test products firsthand, offering a tactile experience that online platforms cannot replicate. This hands-on approach is crucial in aiding consumers' decision-making processes when it comes to electronic products, which often require a feel for usability and quality.

The role of product or sales advisors in customer education and engagement is central to the retail shopping experience and cannot be overstated. With extensive training on product specifications, usage, reviews, and compatibility, product advisors can provide tailored advice that aligns with the specific needs and preferences of each customer (AIContentfy, 2023). This personalized interaction not only enhances customer satisfaction but also fosters loyalty. As consumer electronics and appliances continue to advance technologically, the expertise required by product advisors deepens, necessitating continuous training to keep up with the latest technological advancements and product releases. But with vast amounts of features and functions, to promote the products effectively and accurately to the potential consumer, it is always a great challenge.

Based on the material we have studied; many electronic retail stores are integrating advanced technologies such as augmented reality (AR) and virtual reality (VR) besides personalized service bots to create immersive experiences that simulate product usage (Kim et al., 2022). These technologies help to bridge the gap between a customer's perception and the actual product functionality, thereby aiding in informed purchasing decisions. As consumers today are more informed and hold higher expectations due to easy access to product information online, retail stores are challenged to provide added value through exceptional customer service and expert advice, areas where product advisors are pivotal. This shift in consumer expectations underscores the ongoing relevance of retail stores and their product advisors in the consumer journey, despite the growing prevalence of e-commerce in the electronic consumer industry.

B. Needs of the Consumer Electronics Retail Industry

In the fast-paced consumer electronics retail industry, addressing specific operational and strategic needs is crucial for sustaining growth and enhancing customer satisfaction. To stay competitive, retailers must continuously adapt by integrating emerging technologies such as AI powered gadgets to their product offerings (Hopkins, M., 2023). This approach not only keeps offerings current but also meets the high expectations of today's tech-savvy consumers.

Enhancing customer experience is another need of the industry, especially in a market where many products have minimal technical differentiation. Providing exemplary customer service, personalized marketing, and comprehensive after-sales support can significantly differentiate a retailer in the consumer's eyes (Varnika Om, 2022). Each interaction needs to be positive and tailored, transforming basic transactions into unique experiences that encourage loyalty and repeat business.

Efficient inventory management is equally vital. Retailers must employ accurate demand forecasting and efficient stock management systems to prevent stock surplus (Netstock, 2023), hence minimizing overhead costs to align with the dynamic consumer demand. Additionally, developing a seamless omnichannel presence is essential (Expert Marketing Advisors, 2023). This approach integrates various shopping channels—online, in-store, and mobile—to deliver a unified customer experience. Consumers expect to interact with retailers across multiple platforms seamlessly, whether it’s making purchases, checking product availability, or processing returns (Cook, B., 2023).

Sustainability is becoming increasingly important as public awareness of environmental issues grows. Retailers are pressured to source eco-friendly products and implement greener business practices, which not only help in building a positive brand image but also comply with regulatory standards (Mahmoud et al., 2022; Cohen, S., 2022; Reddy, K. Pradeep., 2023).

Competitive pricing strategies are imperative in a saturated market (Hackett, T., 2024). Retailers can attract price-sensitive consumers through dynamic pricing, bundled offers, and periodic promotions (PROS, Inc., 2023). Furthermore, with the shift towards online retail, protecting customer data with robust cybersecurity measures is paramount to prevent breaches and maintain consumer trust (Walia, E., 2024).

Lastly, given the rapid technological advancements in the industry, ongoing training and development for staff are crucial. Well-informed employees are better equipped to manage and sell modern consumer electronics, providing informed advice and superior customer service.

With all that being said, having a product or sales assistant chatbot can significantly enhance customer experience, not keeping potential customers waiting and providing poor or old advice, and serve as an invaluable tool for new staff by guiding them in improving sales techniques and customer interactions. Acting as a real-time knowledge base, the chatbot provides immediate answers to product-related questions, sales protocols, and customer handling strategies, which can reduce the learning curve for new employees and boost their confidence by providing consistent, accurate information on demand. Additionally, the chatbot can simulate various customer scenarios, allowing new staff to practice and refine their approach to sales conversations. This aspect ensures they are well-prepared to handle different types of customer interactions effectively. Overall, the integration of a sales assistant chatbot empowers new staff by equipping them with the tools and knowledge needed to excel in their roles from the outset.

C. Problems faced by Industry

One of the problems faced by those in the industry is stiff competition. The consumer electronics sector is characterized by intense competition not just within local markets but on a global scale. Retailers must constantly innovate and differentiate their offerings to capture customer loyalty to reduce churn rate. This includes not only the latest technology but also richer and higher quality services like extended warranties, flexible return policies, and loyalty programs. Companies compete on multiple fronts—product variety, price, technological advancements, and customer service quality. To maintain a competitive edge, retailers must stay ahead of market trends, understand consumer preferences, and effectively leverage digital marketing strategies to reach a broader audience (Anchanto, 2022).

Besides that, technological advancement is a core challenge in the consumer electronics industry due to the speed with which new technologies are developed and introduced to the market. This rapid evolution drives demand for the latest devices but also shortens the lifecycle of products. Retailers and manufacturers often face the risk of unsold inventory becoming obsolete as newer models are launched. Effective inventory management and strategic planning are essential to mitigate these risks. Retailers must balance between offering cutting-edge technology and managing the stock of older models, often through discounting or special promotions to clear out older inventory without incurring significant losses.

Today’s consumers are more informed and demanding than ever before. With easy access to information via the internet, consumers often research extensively before making purchases. They expect high-quality products, competitive pricing, and excellent customer service. Additionally, the rise of smart and connected devices has led to expectations for products that are not only functional but also seamlessly integrate with the consumer’s existing ecosystem of devices. Retailers need to ensure their staff is well-trained and that their service channels are equipped to handle complex queries and provide comprehensive support. Enhancing the customer experience in-store and online is crucial for meeting these expectations and fostering loyalty.

Lastly, price sensitivity is particularly pronounced in the consumer electronics market, where similar products are often available from multiple brands and retailers, leading to price comparisons by consumers. Retailers must strategically price their products to attract price-conscious consumers without entering a downward spiral of price wars that erode profit margins. Developing effective pricing strategies, such as dynamic pricing, promotional discounts, and bundle offers, can help attract customers while managing profitability. Additionally, adding value through customer service, exclusive products, or unique shopping experiences can justify premium pricing and help maintain healthy margins.

Overall, the implementation of a product or sales advisor chatbot serves as an invaluable tool for new staff, it is a trend nowadays to provide improved and high value customer interactions. Acting as a real-time and self-learning generative model, the chatbot provides immediate answers to product-related questions, sales protocols, and customer handling strategies, which can reduce the learning curve for new employees and boost their confidence by providing consistent, accurate information on demand through periodical data crawling and knowledge learning from multiple online resources. Additionally, the chatbot can simulate various customer scenarios, allowing new staff to learn, practice and refine their approach to sales conversations and customer interactions. This aspect ensures they are well-prepared to handle different types of customer interactions effectively. Overall, not only does the chatbot help with improving customer service and support, but it also helps make the outlet stand out from the rest of the competition.

D. Objectives

The primary objective of this project is to enhance the efficiency and knowledge distribution among product advisors within the consumer electronics retail industry.

To achieve this, we plan to develop a specialized chatbot tailored to the needs of electronic retail consumers. This tool will assist product advisors in their daily operations by providing accurate product recommendations to customers. The chatbot will also function as a support system for junior product advisors, empowering them to manage customer inquiries and make informed decisions independently, without the need for constant oversight from senior staff. Furthermore, the chatbot will keep all advisors up to date with the latest product lineups, ensuring that the advice and support they offer to customers are current and relevant. This initiative aims to foster a more efficient, knowledgeable, and responsive service environment, significantly enhancing the quality of customer interactions.

From a technical standpoint, we aim to utilize large language models (LLMs) available on HuggingFace and integrate them with our chatbot system. We will test various LLM models and response generation techniques to identify the optimal combination for our system. Additionally, we plan to gather and preprocess relevant data to enhance the response generation capabilities of the chosen LLM model and technique. By leveraging state-of-the-art NLP technologies and continuous data updates, we aim to create a robust and efficient chatbot that significantly improves the productivity and effectiveness of product advisors in the consumer electronics retail industry.

II. LITERATURE REVIEW

This section is organized into four subsections. Subsection A explores the involvement of AI assistants in the retail industry and other sectors, examining their integration and impact on service efficiency and customer interaction.

Subsection B provides a comprehensive summary and analysis of commonly used large language models (LLMs) in the current AI generation landscape, detailing their functionalities and unique characteristics.

Subsection C highlights various works that have utilized LLM models and techniques in generative AI chatbots for sales and service activities, showcasing practical applications and outcomes in different contexts.

Lastly, Subsection D offers a summary of the past works discussed in Subsection C, synthesizing key findings and insights to provide a cohesive understanding of the advancements and applications of LLMs in generative AI chatbots.

A. AI Assistant in Retail and Other Industries Related Works

Chong et al (2021) examined the integration of AI-chatbots as frontline service agents within the retail and consumer services sectors. Utilizing Social Cognitive Theory (Acxiom Technologies LLP, 2023) to frame their analysis, they propose a three-level classification of AI-chatbot design—anthropomorphic role, appearance, and interactivity—and explore how these elements influence the complementarities of agency. Chong et al. recognize the current implementation challenges and suggest that achieving the full potential of AI-chatbots involves navigating the complexities of agency at each design level. They also develop a research agenda that includes the emotional interface, resolving the proxy agency dilemma, and fostering collective agency to enhance the deployment of AI-chatbots as effective service tools (LaMorte, W., 2022). The study provides a thorough investigation into the functional benefits and efficiencies AI-chatbots offer, while also addressing quality of service challenges and the need for better utilization strategies to deliver high-quality consumer services.

Research done by Pantano et al. (2020) and Ostojić, I., (2024) is anchored on a comprehensive analysis of chatbot patents over the past two decades, demonstrating an increasing trend towards the adoption of sophisticated conversational agents capable of engaging customers through natural language. The findings underscore the significant research and development efforts aimed at enhancing the capabilities of chatbots, particularly in terms of drawing inferences from diverse data sources and providing personalized customer interactions (Mehta, J., 2023). Pantano and Pizzi's work not only highlights the technological advancements within the AI field but also maps out the future directions of online customer assistance, offering valuable insights into the evolving interface between consumers and digital systems in retail settings.

Shankar et al (2018) explored how AI, as a combination of programs, algorithms, systems, and machines demonstrating intelligence, is revolutionizing retailing through enhanced product, service, or solution intelligence (Trend Hunter, 2023). His research highlights how AI applications in retailing are expanding due to the exponential growth of business data, exemplified by large retailers like Walmart which processes immense volumes of transaction data. These AI systems are pivotal in helping retailers optimize both demand and supply elements, enhancing customer relationship management and streamlining supply chain efficiencies (Fergus, S., 2024). Shankar's analysis not only underscores the critical role of AI in modern retailing but also provides a strategic framework for understanding and deploying AI effectively in retail operations.

A study focusing on how AI facilitates automated decision-making with precision and speed based on data analytics, complemented by AI's self-learning capabilities was done (Kaur et al., 2020). The research further highlights the substantial changes brought by digitalization, including the impact of eCommerce giants like Alibaba and Amazon, which have elevated consumer expectations. Kaur et al. provide a detailed examination of the revolutionary developments in retail technologies such as AI, Big Data, and the Internet of Things (IoT), discussing how these technologies enhance customer interactions and operational efficiency. Their work offers a comprehensive understanding of AI's expanding role in retail, suggesting that AI not only improves the customer experience but also optimizes retail management practices (Stalmachova et al., 2022; Buehler, T.Leigh., 2024).

Moore et al (2020) explored the impact of AI digital humans on consumer interactions in retail settings (Coherent Market Insights, 2023). The research is grounded in practice-informed, ethnographic methods, focusing on the integration of AI digital humans during the launch phase of a flagship store's digital kiosk greeter. Their findings provide insights into the novel social consequences and opportunities that arise from consumer interactions with AI in retail environments, revealing how these interactions reshape in-store customer experiences and influence broader shopping practices. Moore et al. not only illuminate the complexities of AI in customer service roles but also discuss the managerial implications of integrating AI to enhance the shopping environment and customer engagement. Their work contributes significantly to the understanding of the evolving relationship between digital technology and customer experience in retail contexts (Grewal et al., 2023).

In a paper by Roy (2022), "Artificial Intelligence in Pharmaceutical Sales & Marketing - A Conceptual Overview”, the focus is on how AI is revolutionizing pharmaceutical sales and marketing strategies. Roy delves into the potential of AI to transform traditional marketing approaches through hyper-personalization and hyper-customization, enabling marketers to target individual doctors with unprecedented precision (Ruparelia, 2022). The research highlights AI's role in enhancing Customer Relationship Management (CRM), pre-call planning, and guided sales, which significantly improve sales outcomes and competitive advantage (Roy, 2022; Hashemi-Pour, 2023). Roy's analysis not only underscores the transformative capabilities of AI in pharmaceutical marketing but also maps out future directions for the field, including the integration of conversational AI, natural language processing, and robot-based process automation.

Besides that, Trivedi and Patel (2020) investigated the effects of artificial intelligence and automation on enhancing sales volume in medium-sized enterprises. Anchored on robust statistical methods including M-estimation, S-estimation, and MM-estimation (Susanti et al., 2014), their research found that online content, product quality, and marketing resources significantly boost sales volume, while competition presents a challenge. Trivedi and Patel’s analysis highlights the nuanced role of AI and automation in improving sales strategies, underscoring the potential of these technologies to transform business operations and competitive dynamics in the market. Their work mapped out the pivotal influences of AI and automation, suggesting strategic implementations for businesses aiming to leverage technology for sales enhancement (Dwivedi et al., 2021).

Oosthuizen et al (2020) contributed by exploring the transformative role of artificial intelligence (AI) within the retail sector (V-Count, 2024). Their research provides a conceptual framework for understanding AI's application across the traditional retail value chain, highlighted its potential to streamline operations, enhance customer engagement, and optimize inventory and supply chain management (Integration, 2023). Oosthuizen et al (2020) discussed how AI technologies can be integrated at various stages of the value chain to enhance efficiency and improve customer satisfaction. The paper underscored the significant impact of AI in redefining retail operations and suggests practical implications for retail managers to leverage AI technologies to gain a competitive advantage. The authors’ framework not only illustrates AI's strategic role in retail but also maps out future directions for research and implementation in the industry.

B. *Current LLM AI Generation Model Landscape*

Table 1: AI Assistant Related Workds Summary

Model	Primary Function	Unique Characteristics
GPT-3 (OpenAI)	Text generation, conversation, content creation	- Capable of generating human-like text across various domains. - Adaptable to a wide range of language tasks.
BERT (Google)	Text analysis, language understanding	- Enhances search engines and improves language understanding. - Primarily used for analysis rather than generation.
T5 (Google)	Text-to-text tasks, translation, summarization, question answering	- Converts all text-based language problems into a text-to-text format. - Versatile for various NLP tasks.
Mistral-7B (Mistral AI)	Text generation, summarization, translation	- Designed to be smaller and more efficient than GPT-3. - Maintains similar performance levels.
Llama-2 (Meta AI)	Text generation, conversational agents, content creation	- Offers enhancements in understanding and generating text. - Aims for higher accuracy in diverse contexts.
Falcon 7B (TII)	Text generation, conversational agents, complex task handling	- Focuses on achieving high performance in language generation tasks. - Features enhanced understanding capabilities.

Based on the analysis presented in Table I, we have summarized the capabilities and usage of commonly employed large language models (LLMs) to understand the current landscape of generative AI. Our project will primarily focus on utilizing Mistral-7B, Llama-2, and Falcon 7B for both experimentation and deployment phases. These models were selected because they are open source and do not incur any tokenization fees. Their open-source nature not only reduces costs but also allows for greater customization and control over the implementation process.

C. *LLM Models and Techniques used In Generative AI Chatbot for Sales and Service Activity Related Works*

Tuan et al (2022) successfully showcased Mistral model's effectiveness in reducing both training time and computational expenses while delivering robust performance, making it a viable option for SMEs looking to integrate advanced AI chatbot technologies without substantial resource allocation (Ngoc, 2024). In their past research, they have successfully enhanced the chatbot functionalities in smart city applications using generative AI, specifically focusing on the Mistral model for small- and medium-sized enterprises (SMEs). Their research, which emphasized the capabilities of the Mistral model (Fernandex, 2024), a variant of the BLOOM language model, showcased how it efficiently automates question-response interactions across multiple languages. Tuan et al highlighted that this study not only underscores the adaptability and efficiency of the Mistral model but also illustrated its potential to significantly enhance customer service and urban management within the smart city framework.

Kim and Min (2024) integrated a fine-tuned Mistral model within the QA-RAG framework and demonstrated significant improvements in accuracy, outperforming conventional models in providing precise regulatory guidance (Kim and Min, 2024). In their paper, they studied the innovative adaptation of the QA-RAG model (LangChain, 2024), leveraging the Mistral model (Fernandex, 2024) to improve pharmaceutical regulatory compliance. Their research is grounded in addressing the intricate and voluminous guidelines that often burden industry professionals (Kim and Min, 2024). Kim and Min's work not only underscores the effectiveness of the Mistral model in navigating complex regulatory environments but also provides a promising avenue for employing advanced AI in critical compliance domains, offering a blueprint for future enhancements in AI-driven regulatory processes.

Bhattacharyya et al (2024) investigated the enhancement of chatbot interfaces in educational institutions. Their research is anchored on the implementation of Large Language Models (LLM) like Mistral-7B and Palm 2, focusing on custom information integration to facilitate non-standard inquiries by users. The study demonstrates that AI-driven chatbots can significantly reduce the need for human interaction by providing instant, accurate responses to institute-related queries, thus alleviating the workload on sales personnel and enhancing user satisfaction. The authors highlight the seamless deployment of these technologies using platforms like AWS and frameworks such as Flask, underscoring the practical benefits of AI in improving administrative efficiency and decision-making processes in educational settings (MoldStud, 2024). Their work not only showcases the adaptability and efficiency of using advanced LLMs in chatbot development but also illustrates its potential to significantly enhance customer service and operational management in educational contexts (Ali et al., 2024).

Dhake (2024) analyzed the transformative impact **of** Generative AI (GenAI) and Large Language Models (LLM) within the banking sector, emphasizing their role in enhancing customer service and optimizing back-office operations. The paper discusses the integration of these technologies into banking infrastructures to automate tasks, enhance security, and personalize customer interactions (Božić, 2023). It also addresses challenges such as ethical considerations, AI transparency, and data sensitivity (Elsaid, 2024). Dhake et al.'s (2024) study is significant for providing insights into responsibly deploying GenAI and LLMs in financial services, highlighting the importance of robust regulatory frameworks to ensure fairness and privacy.

Dhoni (2024) provided an in-depth analysis of artificial intelligence applications in the retail sector, with a particular focus on how machine learning and generative AI technologies are transforming the landscape of customer experience and operational efficiency. The authors comprehensively discuss various AI implementations, such as the optimization of inventory management through predictive algorithms and the personalization of marketing strategies via data-driven insights (Hypersonix & Becchetti, 2024). Furthermore, the paper delves into the challenges associated with these technologies, including concerns over data privacy, the financial implications of AI deployment, and the requisite expertise needed to manage these sophisticated tools (Elsaid, 2024). A significant part of the study emphasizes the need for ethical considerations in AI deployment, highlighting the potential risks and mitigation strategies to prevent data misuse and ensure consumer protection. Dhoni et al.'s work serves as a pivotal resource for retailers considering AI integration, offering a nuanced view of the opportunities for enhancing service delivery alongside the operational hurdles that must be navigated.

Ding et al (2024) investigated the transformative potential of AI in the educational sector, focusing on how these technologies tailor learning experiences to meet individual student. The paper reviews a range of AI-powered tools, from automated grading systems to AI-driven tutoring that can adapt in real time to the learning progress of students (Joadekar, 2024). It explores the integration of AI in creating dynamic educational content that not only engages students but also supports diverse learning styles and speeds. Despite the promising applications, the authors critically address the challenges of relying heavily on AI, such as the potential loss of critical human interaction and the risk of widening educational disparities if access to these technologies is uneven (OnFinance AI, 2024). The study calls for a balanced approach to technology integration, where AI complements traditional teaching methods, ensuring that educational technologies are inclusive and effectively enhance teaching and learning processes.

Gao et al (2024) explored the enhancement of large language models (LLMs) for marketing analytics, particularly focusing on the open-source LLM, Llama-2-70b. They investigate how techniques such as semantic search and fine-tuning can be applied to improve Llama-2-70b's capabilities in complex tasks like SQL generation and tabular data analysis, crucial for precise marketing decision-making. By embedding documents as numerical vectors and aligning queries with relevant information, semantic search significantly enhances Llama-2-70b's contextual understanding. Further fine-tuning on domain-specific datasets allows the model to excel in specialized tasks traditionally challenging for LLMs (Pandey, 2023; Rizvi, 2023). Gao et al.'s comparative analysis demonstrates Llama-2-70b's competitive or even superior performance against proprietary models, highlighting its potential to transform marketing analytics through advanced, open-source AI technology.

D. Analysis of LLM Models and Techniques used in Generative AI Chatbot for Sales and Service Activity Related Works

Table 2: Generative AI Related Works Summary

Author	Problems	Techniques Used	Contributions
Tuan et al. (2022)	Investigating AI's role in retail from a managerial perspective, focusing on ethical and operational aspects.	Utilization of the Mistral model, a variant of the BLOOM language model, to automate question-response interactions in multiple languages.	- Reduced training time and costs with the Mistral model. - Enhanced chatbot functions in smart city environments. - Showcased adaptability and benefits for customer service and urban management.
Kim and Min (2024)	Challenges in navigating complex regulatory environments in the pharmaceutical industry using conventional models.	Integration of the fine-tuned Mistral model within the QA-RAG framework to enhance pharmaceutical regulatory compliance.	- Highlighted Mistral model's superior accuracy in providing precise regulatory guidance. - Emphasized its strategic role in AI compliance. - Suggested prospects for future advancements in AI.

Bhattacharyya et al. (2024)	Need to reduce human interaction in educational institutions and improve response times for institute-related queries.	Implementation of Large Language Models (LLM) like Mistral 7B and Palm 2, integration of custom information to handle non-standard inquiries.	- Demonstrated AI-driven chatbots' role in reducing workload and enhancing user satisfaction. - Highlighted efficient deployment using AWS and Flask. - Showcased improvements in administrative efficiency and decision-making in educational settings.
Dhake, S. (2024)	Integration of Generative AI and Large Language Models in the banking sector, focusing on regulatory, ethical, and data security issues, along with the need to balance automation with personalized interactions.	Utilization of Generative AI for automation, Large Language Models for enhancing customer service, and the development of ethical frameworks to guide AI deployment.	- Analyzed the impact of generative AI and large language models in banking. - Outlined strategies for ethical integration. - Addressed regulatory compliance issues.
Dhoni, P. S. (2024)	Challenges in optimizing retail operations and enhancing customer experiences with AI, focusing on data privacy, cost, and the technical expertise required for AI implementation.	Machine learning algorithms for inventory management, generative AI for marketing personalization, ethical frameworks for data protection.	- Provided a comprehensive overview of AI applications in retail. - Evaluated operational and ethical challenges. - Recommended strategies for AI deployment.
Ding, M., Dong, S., & Grewal, R. (2024)	Enhancing educational experiences through personalized learning and the challenges of integrating AI without diminishing the role of human educators.	AI-driven adaptive learning systems, automated grading tools, real-time feedback mechanisms.	- Offered insights on effective AI integration in education. - Discussed balancing AI tools with human teaching methods.
Gao, Y., Arava, S. K., Li, Y., & Jr, J. Ws. (2024)	Challenges in using standard LLMs for SQL generation and tabular data analysis in marketing analytics, highlighting the need for enhanced precision and applicability for specialized tasks.	Llama-2-70b and other Large Language Models, semantic search techniques, fine-tuning on domain-specific datasets.	- Conducted a detailed evaluation of Llama-2-70b and other LLMs in marketing analytics. - Presented methodological advancements to enhance LLM functionality in complex analytical tasks.

Table II provides a consolidated overview of recent research on the deployment and effectiveness of large language models and generative AI in various professional domains. It highlights key findings from multiple studies that explore the transformative potential of these technologies in enhancing operational efficiencies, decision-making processes, and user satisfaction across sectors such as smart city management, banking, education, and retail.

III. METHODOLOGY

In this paper, we detail our research process for utilizing Falcon7B and Llama2 models. Section A outlines the research flow, including data preparation, model training and evaluation, and chatbot interactions. Sections B and C focus on the operational aspects and setup requirements of Falcon-7B and Llama2. Finally, Section D offers a concise summary and evaluation of the chapter, tying together all the key aspects of our research methodology.

A. *Flow of Research Activities*

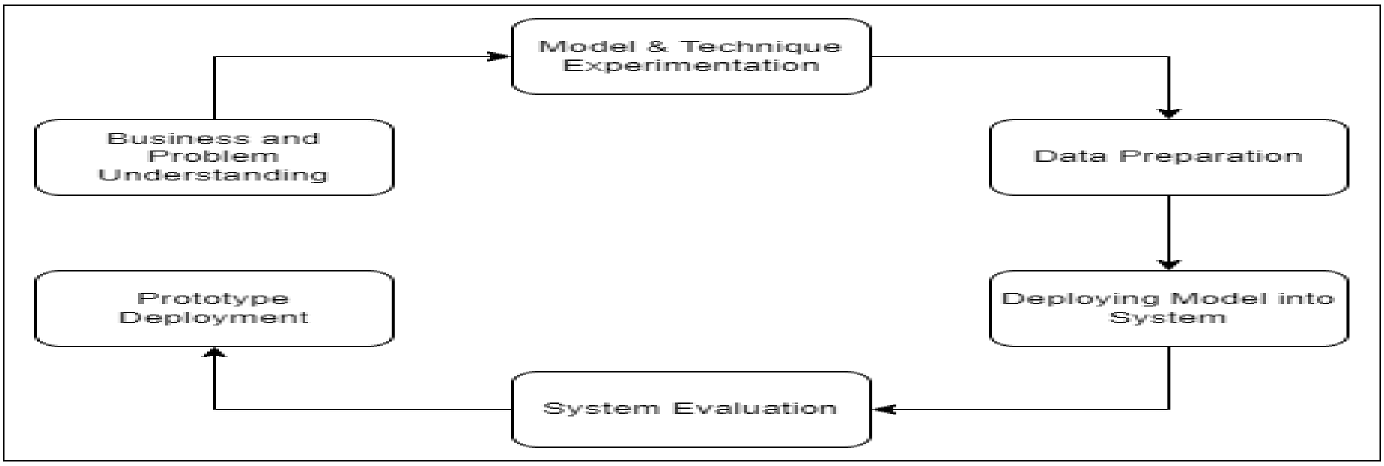

Model & Technique Experimentation
Data Preparation
Deploying Model into System
System Evaluation
Prototype Deployment
Business and Problem Understanding

Fig 1: Flow of Research Activites in the Project

In Fig. 1, the research activities for developing an AI chatbot are divided into six distinct phases, each crucial to the project's success.

In the initial phase, Business and Problem Understanding, we conduct a thorough analysis of the challenges and needs within the Consumer Electronics Retail Industry. This phase is essential as it helps us align our project objectives with the industry's demands. By understanding the specific problems faced by this sector, we can ensure our AI chatbot will provide meaningful and relevant solutions.

The second phase, Model and Technique Experimentations, involves exploring various open-source Large Language Models (LLMs) and inference techniques. During this phase, we evaluate models such as Llama2, Falcon-7B, and Mistral-7B. Additionally, we assess inference techniques like Retrieval Augmented Generation (RAG), Fine-Tuning using QLoRA, and LLM Chaining. This experimentation is critical for identifying the most effective models and techniques for our specific application.

Following our experimentation, we move on to the Model Selection and Data Preparation phase. Based on the results of our evaluations, we select the final model and techniques. Subsequently, we prepare the necessary data, ensuring it is clean, relevant, and properly formatted to support the chosen model. This phase is foundational, as it lays the groundwork for the subsequent development stages.

In the System and Model Development phase, we create a robust pipeline to integrate the selected LLM with any additional features required by the system. This involves setting up processes for data input, model inference, and output generation, ensuring seamless operation. Proper pipeline development is crucial for maintaining the system's efficiency and reliability.

Once the system is established, we proceed to the System Evaluation phase. During this phase, we rigorously test the chatbot to assess its performance, accuracy, and practicality. This evaluation helps us determine whether the system meets the desired standards and is ready for deployment. It also allows us to identify and address any potential issues before the system goes live.

The final phase, deployment, involves deploying the chatbot through user-friendly platforms such as Streamlit. This step enables us to present and demonstrate the system to users, gather feedback, and make any necessary adjustments to improve functionality and user experience. By following these structured phases, we ensure a comprehensive and methodical approach to developing an AI chatbot tailored to the needs of the Consumer Electronics Retail Industry.

B. *Flow of Falcon-7B*

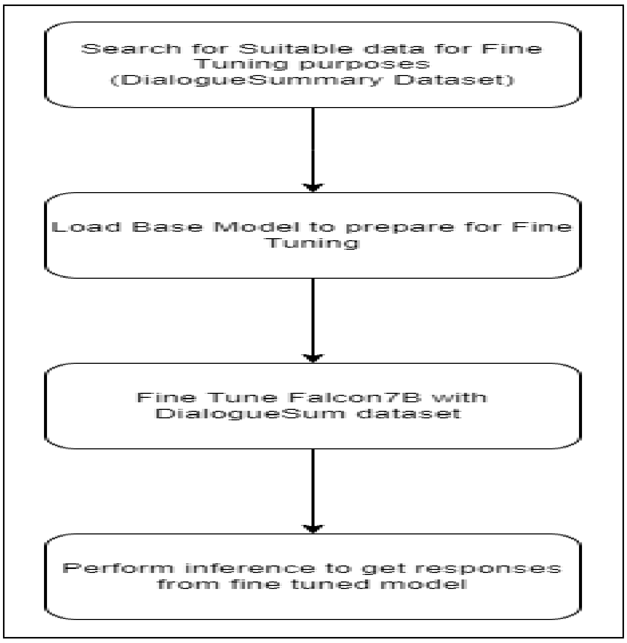

Search for Suitable data for Fine Tuning purposes (DialogueSummary Dataset)
Load Base Model to prepare for Fine Tuning
Fine Tune Falcon7B with DialogueSum dataset
Perform inference to get responses from fine tuned model

Fig 2: Flowchart on the Preparation of Falcon-7B

To fine-tune a conversational model using data from Hugging Face with a base model like Sharded Falcon 7B, a detailed and methodical approach is essential. Referring to Fig. 2, the process begins with the acquisition of suitable conversational data. For this purpose, the 'Dialogue Sum' dataset available on Hugging Face is an excellent choice. It is specifically tailored for dialogue systems and offers a rich source of conversational exchanges.

Once the data is obtained, the crucial step of preprocessing comes into play. This stage involves cleaning and formatting the data to make it suitable for training the model. This process includes removing irrelevant or redundant information, fixing any errors in the format, and possibly breaking down longer dialogues into smaller, more manageable parts. The objective here is to ensure that the dataset is consistent, relevant, and structured in a way that the model can easily process and learn from.

With the dataset prepared, the focus shifts to the base model, in this case, the Sharded Falcon 7B. This model is known for its large scale, efficiency, and ability to handle extensive datasets. Loading the Shared Falcon 7B involves setting up the necessary computational resources and initializing the model with its pre-trained parameters. This step is critical as it lays the foundation for the fine-tuning process.

The actual fine-tuning of the model is conducted with the prepared dataset, which, in this scenario, includes about 5,000 rows of conversational data. This stage is where the model learns from the specific nuances and contexts of the dataset, adapting its pre-trained knowledge to the specificities of the conversational style and content in the 'Dialogue Sum' data. This process is both resource-intensive and time-consuming, often taking around 8 hours to complete.

After fine-tuning, it's essential to evaluate the model's performance. This is achieved through a process called inference, where new, unseen data is fed into the model to assess how well it responds. The quality and relevance of the model's responses are key indicators of its learning and adaptation capabilities. This evaluation helps in understanding how effectively the model can handle real-world conversational scenarios based on the training it received.

Throughout this entire process, from data acquisition to model evaluation, it is vital to continuously monitor the model's performance and make necessary adjustments. This iterative approach ensures that the model is fine-tuned to the highest efficiency, making it capable of handling complex conversational tasks in a wide range of applications.

C. *Flow of Llama2 + Langchain*

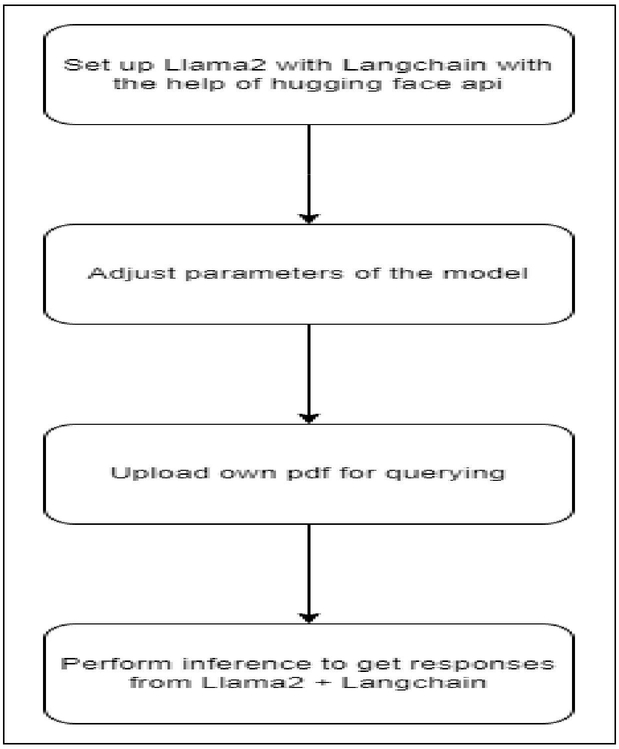

Set up Llama2 with Langchain with the help of hugging face api
Adjust parameters of the model
Upload own pdf for querying
Perform inference to get responses from Llama2 + Langchain

Fig 3: Flowchart on the Preparation of Llama2 + Langchain

From Fig. 3 above, the first step in this process involves importing all the necessary libraries. This typically includes libraries for handling PDF files, such as PyPDF2 or pdfplumber, which are used for reading and extracting text from PDF documents. Additionally, libraries for data processing and machine learning tasks, such as numpy for numerical operations, pandas for data manipulation, and torch for deep learning, might also be required. Importing these libraries at the beginning ensures that all the necessary functions and methods are readily available for use in subsequent steps.

Once the libraries are imported, the next task is to load the PDF data. This involves reading the PDF file and extracting the text contained within it. PDF files can be complex, with text distributed across multiple pages, sometimes in non-linear formats. Therefore, careful handling is required to ensure that the text is extracted accurately and comprehensively. The extracted text is then typically split into manageable chunks. This is crucial for processing, as very long texts can be challenging to work with directly. Splitting the text into smaller parts makes it more manageable and allows for more efficient processing in later stages.

The following stage involves creating embeddings for each text chunk. This is where Pinecone, an embedding database service, comes into play. Pinecone is initialized and then used to convert the text chunks into embeddings. These embeddings are numerical representations of the text data, capturing the semantic meaning of the text in a format that can be easily processed by machine learning models. The creation of embeddings is a vital step, as it transforms the raw text into a form that is suitable for inference and analysis.

The final step is to load the Llama2 model with LangChain for performing inferences. LangChain is a toolkit that facilitates language model operations, making it easier to work with models like Llama2. The Llama2 model, once loaded, is then used to perform inferences on the text embeddings. This might involve tasks like text classification, sentiment analysis, or other forms of natural language processing, depending on the specific requirements of your project.

D. Hardware and Software Used

This section outlines the software and tools integral to this project, which facilitate our experimentation and the development of the chatbot system. Through these resources, we are equipped to advance our project objectives efficiently.

➢ *NVIDIA RTX 3090 GPU*

This hardware component is critical for training and running large language models like the ones used in LLM chatbots due to their ability to process large datasets and perform complex computations rapidly. GPUs significantly accelerate the training and inference phases of machine learning models.

➢ *Jupyter Notebook*

It is an interactive computing environment where we can write, test, and debug the chatbot’s code live. It is particularly useful for experimenting with and fine-tuning the LLM models within the notebook.

➢ *Hugging Face*

Hugging Face is a platform that offers downloading, configuring, and deploying state-of-the-art pre-trained models, including LLMs. We have used it to access models like Falcon-7B, Llama2 and Mistral 7B, enabling us to access these open-source models easily.

➢ *Docker*

Docker is used to containerize the environment where the LLM chatbot operates. This ensures that the chatbot application runs consistently across different computing environments by encapsulating it along with its dependencies. Containerization with Docker simplifies deployment, scales seamlessly across different machines, and isolates the application, making it more secure.

IV. DATA AND PRELIMIINARY WORKS

In this section, we delve into data preparation, preprocessing, and experimentation methodologies in conversational AI. Section A covers data preparation, detailing our data sources for the experiments, including data size, type, and the preprocessing techniques employed.

Section B examines the general architecture of conversational generative models. It provides an in-depth analysis of the Falcon7B model and explores the architectures of Llama2 and Langchain.

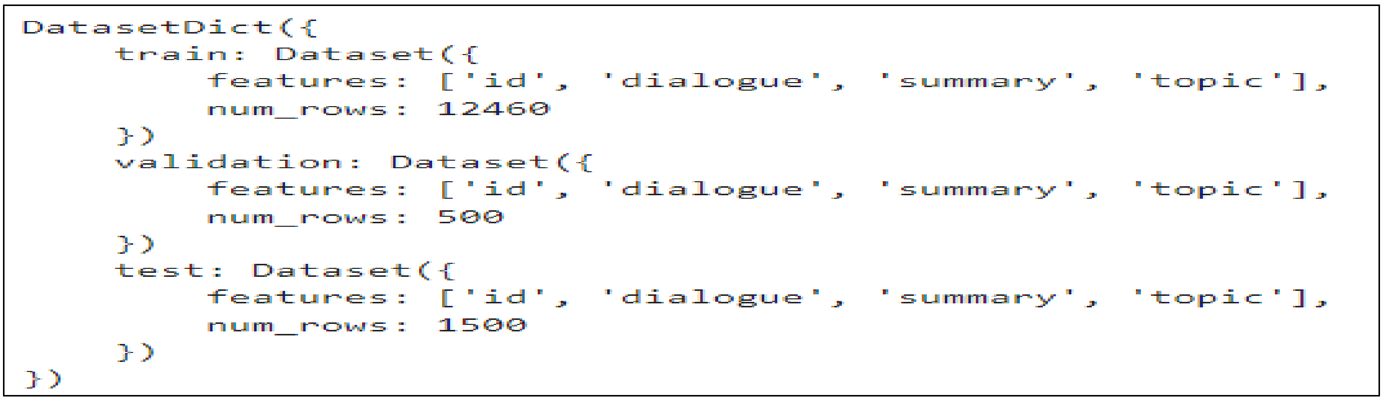
Section C documents the experimental processes, focusing on the experimentation with Falcon7B and the combined application of Llama2 and Langchain.

A. Data Preparation

➢ *Data Source*

The dataset "knkarthick/dialogsum" from Hugging Face is integral to the training of conversational and generative AI. This repository encompasses a spectrum of dialogues drawn from everyday scenarios, reflecting the multifaceted nature of human interactions. The dialogues span various sectors including education, healthcare, and commerce, and feature a range of social interactions, from casual conversations to formal service encounters. This extensive and varied dataset is critical for the development of AI systems capable of replicating human conversational patterns across a breadth of topics and situational contexts, which is essential for their application in diverse real-world environments. The richness of the dataset provides a robust platform for AI to learn nuanced communication, thereby enhancing their ability to engage in and sustain contextually relevant and socially aware dialogues.

➢ *Data Size*



```
DatasetDict({
    train: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 12460
    })
    validation: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 500
    })
    test: Dataset({
        features: ['id', 'dialogue', 'summary', 'topic'],
        num_rows: 1500
    })
})
```

Fig 4: Details of “knkarthick/dialogsum” dataset

The "knkarthick/dialogsum" dataset is composed of a total of 14,460 dialogue entries, each accompanied by expertly crafted summaries and assigned topics as shown in Fig. 4. The dataset is structured into three subsets: 12,460 entries for training purposes, 500 entries designated for validation, and 1,500 entries reserved for testing. This organization supports the systematic training, validation, and assessment of AI models in natural language processing tasks.

- *Data Type*

The "knkarthick/dialogsum” dataset consists of four columns: id, dialogue, summary, and topic. A detailed explanation of each column is listed in Table III below.

Table 3: Data Description

Column	Description
Id	Unique Identifier
Dialogue	Conversation between two persons
Summary	A summary of conversation between two persons
Topic	The topic of the conversation

- *Data Preprocessing*

For Falcon7B, we will have to preprocess the data to fine-tune it further with our desired dataset. However, for Llama2, no fine-tuning is being performed hence we will not need to preprocess any data for Llama2. We will refer to Appendix A for an in-depth discussion of the code used to prepare the data.

B. Architecture of Generative Models

- *Falcon-7B*

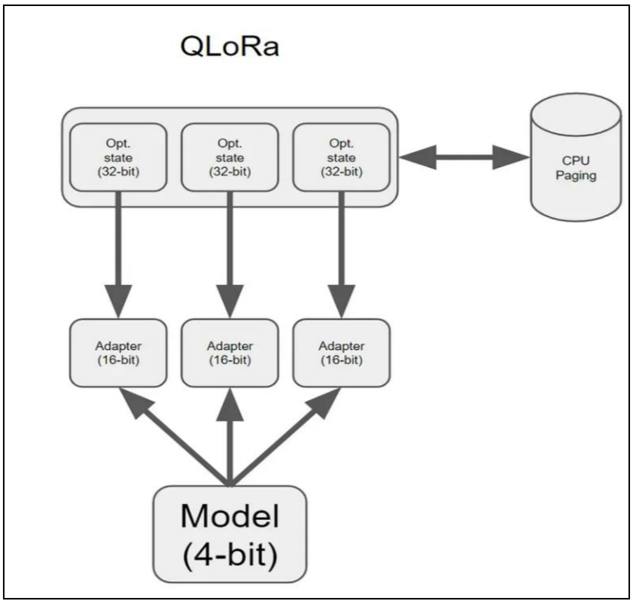


Fig 5: Architecture of Falcon-7B

By referring to Fig. 5, the model that is trained using QLoRa prioritizes quantization, which is the process of constraining an input from a large set to output in a smaller set. In neural networks, quantization typically pertains to reducing the precision of the numbers used to represent the model's parameters, which can drastically reduce the model's memory footprint and speed up computation.

In this hierarchical setup, the topmost layer comprises multiple 32-bit optimized state units, which represent a high-resolution state of the system's parameters or data handling units. These units could be responsible for initial data processing or retention of high-precision computations that are necessary before quantization. The connection to CPU means that it is responsible for managing memory efficiently, an important aspect for high-performance computing and real-time analytics.

The middle tier consists of 16-bit adapters, which act as intermediaries. These adapters are used to downscale the data from a higher to a lower precision, making it more manageable for the model at the base of the architecture. The use of adapters is a modular approach, where data or computational precision can be adjusted as needed, potentially allowing for dynamic scaling based on the demands of the task or the limitations of the hardware.

The base of the architecture is a 4-bit model, which is considered extreme quantization. Models with such low bit-widths are highly unusual but might be employed in extremely resource-constrained environments or for very rapid processing where precision is less critical. This shows that Falcon7B is highly optimized for speed and resource efficiency, possibly at the expense of some accuracy or resolution in its computations.

The flow from 32-bit to 4-bit is a gradual reduction and refinement process, where data or computations are streamlined and optimized at each level to achieve a balance between performance and resource usage. This could be particularly beneficial in applications like mobile devices, embedded systems, or IoT devices, where computational resources are limited, but a fast response time is essential.

- *Llama2 + Langchain*

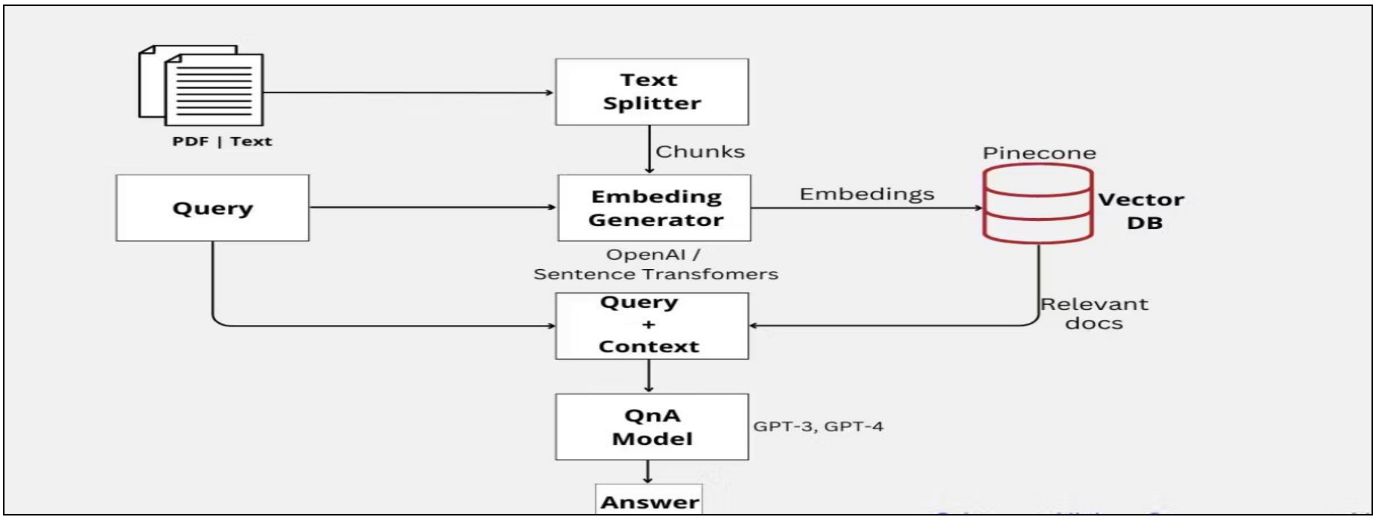


Fig 6: Architectural Flowchat of Llama2 + Langchain

Based on Fig. 6, the system begins with the input of textual data, typically in the form of PDFs or digital text documents. This data is meticulously segmented into smaller, manageable chunks by a dedicated Text Splitter. The segmentation is crucial to facilitate detailed analysis and to ensure that the subsequent processes can handle the data efficiently.

Each text chunk is then transformed into a vector representation through an Embedding Generator. This step involves using sophisticated AI algorithms to translate text into numerical forms that capture the essence of the words and their contextual meanings. The generated embeddings are indexed and stored in a specialized Vector Database, such as Pinecone, designed for high-performance similarity searches.

When a query is received, it is not processed in isolation. Instead, it is enriched with context from the Vector Database, ensuring that the system's understanding is as nuanced and detailed as possible. The Llama2 model, known for its advanced natural language processing capabilities, leverages this rich context to analyze and interpret the query within the appropriate frame of reference.

The final step is where the Llama2 model synthesizes the insights gleaned from the query and its context to generate a precise and relevant answer. This model, being at the cutting edge of AI technology, is adept at crafting responses that are not only accurate but also coherently structured, reflecting a deep understanding of the subject matter at hand.

C. Experiments Carried Out

- *Falcon-7B*

The Falcon 7B model represents a significant milestone in the advancement of generative AI, bringing together state-of-the-art technology and sophisticated algorithms to create a powerful tool for natural language processing and generation. Falcon 7B, a part of the broader Falcon series of language models, is known for its exceptional capabilities in understanding, interpreting, and generating human-like text. It stands out due to its large-scale architecture, which allows it to process and analyze vast amounts of data with remarkable accuracy and depth.

Falcon 7B's design is geared towards handling a wide array of language-related tasks. These include but are not limited to generating coherent and contextually relevant text, answering complex queries, and providing insightful analyses. Its ability to understand and generate natural language makes it an invaluable asset in various applications, such as automated content creation, conversational AI systems, and sophisticated data interpretation tasks.

One of the key strengths of Falcon 7B is its versatility and adaptability. It can be fine-tuned for specific applications, enabling it to deliver highly tailored responses and analyses. This customization makes it particularly useful in sectors where nuanced understanding and specialized knowledge are crucial, such as in healthcare, finance, and legal industries.

Moreover, Falcon 7B's advanced capabilities in generative AI are complemented by its user-friendly interface and integration potential. It can be seamlessly incorporated into existing systems and workflows, making it accessible not only to AI researchers and data scientists but also to businesses and organizations looking to leverage the power of AI for enhancing their operations.

In summary, the Falcon 7B model is a robust, scalable, and versatile tool in the field of generative AI. Its sophisticated architecture and advanced language processing capabilities make it a cutting-edge solution for a range of applications, driving innovation and efficiency across various industries. The Falcon 7B model stands as a testament to the rapid progress in AI technology, heralding a new era of intelligent systems capable of understanding and interacting with human language in unprecedented ways.

In Appendix B of this paper, we discuss about how data is being setup and preprocessed for fine-tuning. This is followed by Appendix C where we will demonstrate the code used to inference the fine-tuned model together with experimentation with RAG technique on pdf document.

➢ *Llama2 + Langchain*

Llama2 with LangChain represents a cutting-edge advancement in the field of generative AI, combining the power of a sophisticated language model with the efficiency and flexibility of a specialized toolkit. LLaMA2, standing as a part of the Llama (Large Language Model - Atlas) family, is a highly advanced language model known for its exceptional language understanding and generation capabilities. This model is designed to handle a wide range of natural language processing tasks, making it an invaluable tool for generating human-like text, understanding context, and providing relevant, accurate responses to a variety of prompts.

LangChain, on the other hand, serves as a complementary toolkit specifically designed to enhance the performance and usability of large language models like LLaMA2. It simplifies the process of integrating these models into applications, providing a streamlined and user-friendly interface. LangChain allows for efficient handling of model operations, including loading models, preprocessing text data, and managing model inferences. This integration significantly reduces the complexity and technical overhead typically associated with deploying large-scale language models, making it more accessible for a broader range of users and applications.

The combination of Llama2 and LangChain in the realm of generative AI opens a plethora of possibilities. From creating compelling and coherent narratives to answering complex queries with nuanced understanding, this duo is well-equipped to tackle diverse challenges in the AI space. Their application extends across various domains, including but not limited to, content creation, conversational AI, information extraction, and automated text summarization.

The use of Llama2 with LangChain in generative AI not only enhances the quality of the generated content but also ensures a smoother and more efficient workflow for developers and AI practitioners. This integration marks a significant step forward in the journey towards more advanced, user-friendly, and capable AI systems, paving the way for innovative applications that were once thought to be beyond the reach of automated systems. We will refer to Appendix D and E to explore the process of setting up Llama2 with Langchain together with performing RAG on the pdf document to observe the response.

Overall, in comparison to Falcon7B, Llama2 will give a better response as it is suitable for pdf querying tasks, however, if we wish to have a more human-like response like ChatGPT, Falcon7B holds the potential to achieve this goal as long as we have sufficient and relevant data for the subject requirements.

V. TECHNIQUES CONSIDERATION & OVERALL MODEL DEVELOPMENT

A. *Techniques Consideration*

After experimenting with fine-tuning and RAG techniques for the generation of the model responses, we have concluded that using RAG is more efficient in contrast to fine-tuning due to our limited resources. Instead of fine-tuning periodically, we will find quality data and update the dataset used for RAG. This will not only save time but also improve the model’s response generation overall.

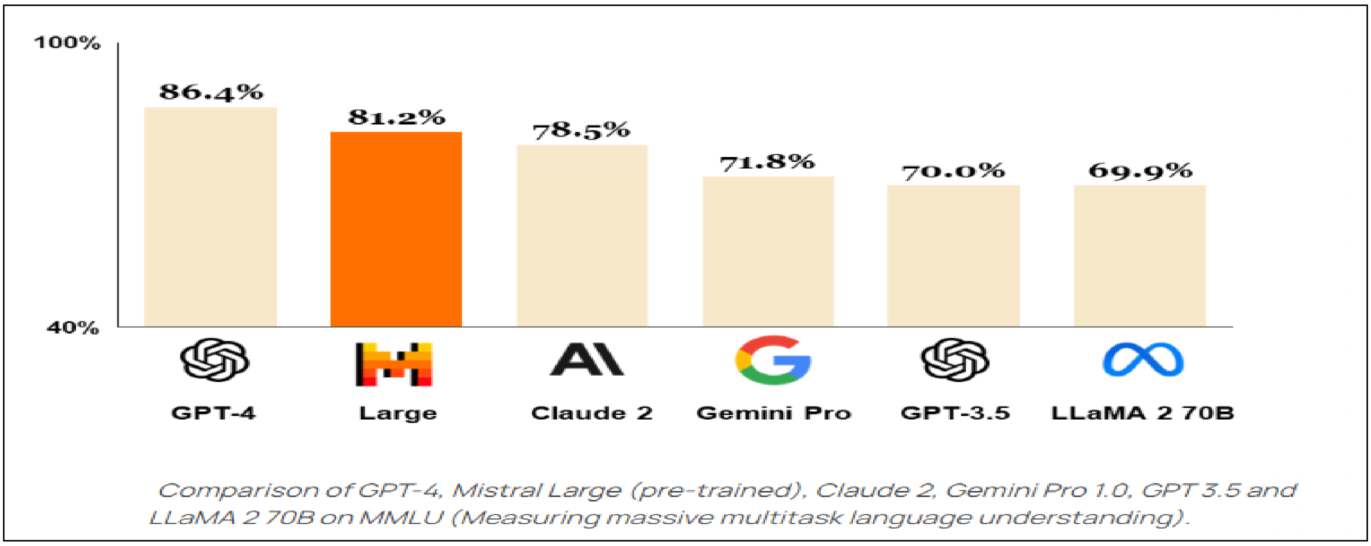


Fig 7: Model Performance Benchmark Table from mistral.ai

We have chosen to use a pretrained Mistral-7B model as our final model for the chatbot system as it has the best performance overall when compared to models like Llama2, although it performs similar with Falcon7B, the data that has been trained on Mistral is more up to date. Hence the selection on why we chose Mistral-7B. The comparison of results between the models can be shown in Fig. 7 above.

In the development of the Retrieval-Augmented Generation (RAG) component of our LLM-based chatbot, we specifically tailored the dataset to enhance the model's ability to provide relevant and informed responses related to mobile phone reviews. Here is a detailed breakdown of the data collection and preparation process:

➢ *Data Collection Strategy*

Initially, we compiled a comprehensive list of smartphone offerings from a local consumer electronic product chain store. This list served as the foundation for our targeted data scraping, ensuring that the reviews gathered were relevant to the current products in the market.

Subsequently, we turned to GSMArena to scrape expert reviews for each smartphone listed. It was crucial to adhere to ethical scraping practices; hence, we strictly complied with the directives outlined in GSMArena's robots.txt file. This compliance ensured that our data collection methods were respectful of the website's terms of service and did not disrupt its normal operation.

➢ *Data Scraping Implementation*

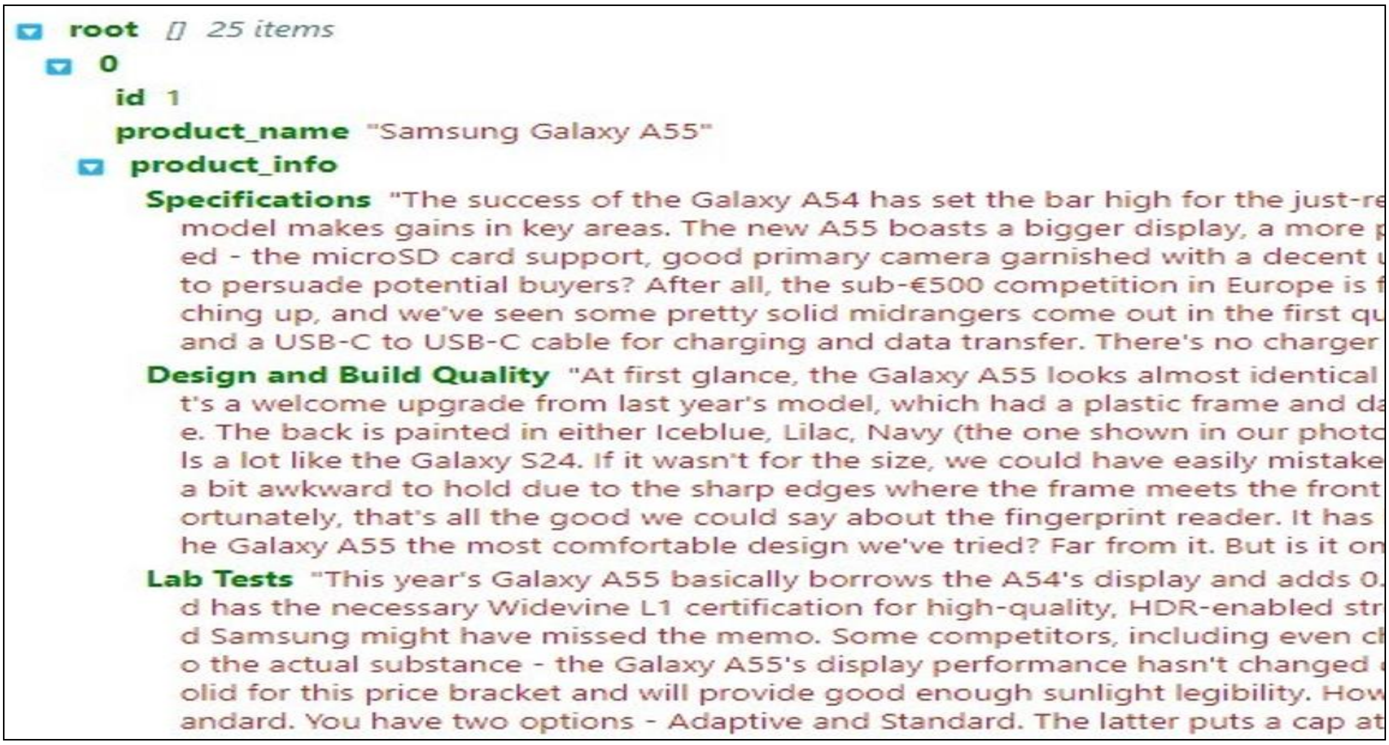
To efficiently scrape the reviews without causing undue strain on GSMArena's servers, we implemented a methodical scraping process. We incorporated the sleep() function in our scraping script, introducing deliberate pauses between consecutive requests. This approach minimized the risk of overwhelming the server with high-frequency requests, which can lead to IP bans or other restrictive measures against our scraping activities. The code to the scraping implementation can be referred at Appendix F.

The scraping operation was conducted over a span of approximately two hours, during which we systematically collected reviews for all smartphones offered by a local consumer electronic product chain store listed on their site.

➢ *Data Storage and Preprocessing*

Once collected, the review data was formatted and stored in JSON Lines format (JSONL). This format was chosen due to its efficacy in handling large datasets where each new record is stored as a separate line. JSONL is particularly advantageous for NLP tasks as it facilitates easy access to each data point without the need to load the entire dataset into memory.

The stored JSONL files then underwent preprocessing to structure the review data effectively for the RAG model. This preprocessing involved cleaning the text data, such as removing unnecessary formatting, correcting typos, and standardizing expressions. This step was critical to ensure that the input data was clean, well-structured, and conducive for training the model efficiently.



```
root  [] 25 items
  0
    id 1
    product_name "Samsung Galaxy A55"
    product_info
      Specifications "The success of the Galaxy A54 has set the bar high for the just-re
        model makes gains in key areas. The new A55 boasts a bigger display, a more p
        ed - the microSD card support, good primary camera garnished with a decent u
        to persuade potential buyers? After all, the sub-€500 competition in Europe is f
        ching up, and we've seen some pretty solid midrangers come out in the first qu
        and a USB-C to USB-C cable for charging and data transfer. There's no charger
      Design and Build Quality "At first glance, the Galaxy A55 looks almost identical
        t's a welcome upgrade from last year's model, which had a plastic frame and da
        e. The back is painted in either Iceblue, Lilac, Navy (the one shown in our photo
        ls a lot like the Galaxy S24. If it wasn't for the size, we could have easily mistake
        a bit awkward to hold due to the sharp edges where the frame meets the front
        ortunately, that's all the good we could say about the fingerprint reader. It has
        he Galaxy A55 the most comfortable design we've tried? Far from it. But is it on
      Lab Tests "This year's Galaxy A55 basically borrows the A54's display and adds 0.
        d has the necessary Widevine L1 certification for high-quality, HDR-enabled str
        d Samsung might have missed the memo. Some competitors, including even ch
        o the actual substance - the Galaxy A55's display performance hasn't changed
        olid for this price bracket and will provide good enough sunlight legibility. How
        andard. You have two options - Adaptive and Standard. The latter puts a cap at
```

Fig 8: Example of Review Data

Here is an example of the data in a jsonl format shown in Fig. 8.

The dataset for the RAG (Retrieval-Augmented Generation) model is meticulously organized into three columns: id, product_name, and product_info. Each element of this structure plays a critical role in optimizing the model's functionality and performance.

The id column assigns a unique identifier to each entry, ensuring that each product review can be distinctly referenced and accessed. In the context of RAG, where retrieved content needs to be linked back to a source or used in a sequential process, having a unique id for each data point simplifies tracking and managing data throughout the retrieval and generation phases.

The product_name column, which contains the name of the smartphone, acts as a crucial descriptor that aids the RAG model in matching queries with the relevant product reviews. By explicitly incorporating the product name, the model can efficiently filter and retrieve information directly related to specific user queries about products. This feature is pivotal in enhancing the user experience, as it ensures that the responses generated by the chatbot are pertinent and tailored to inquiries about specific models.

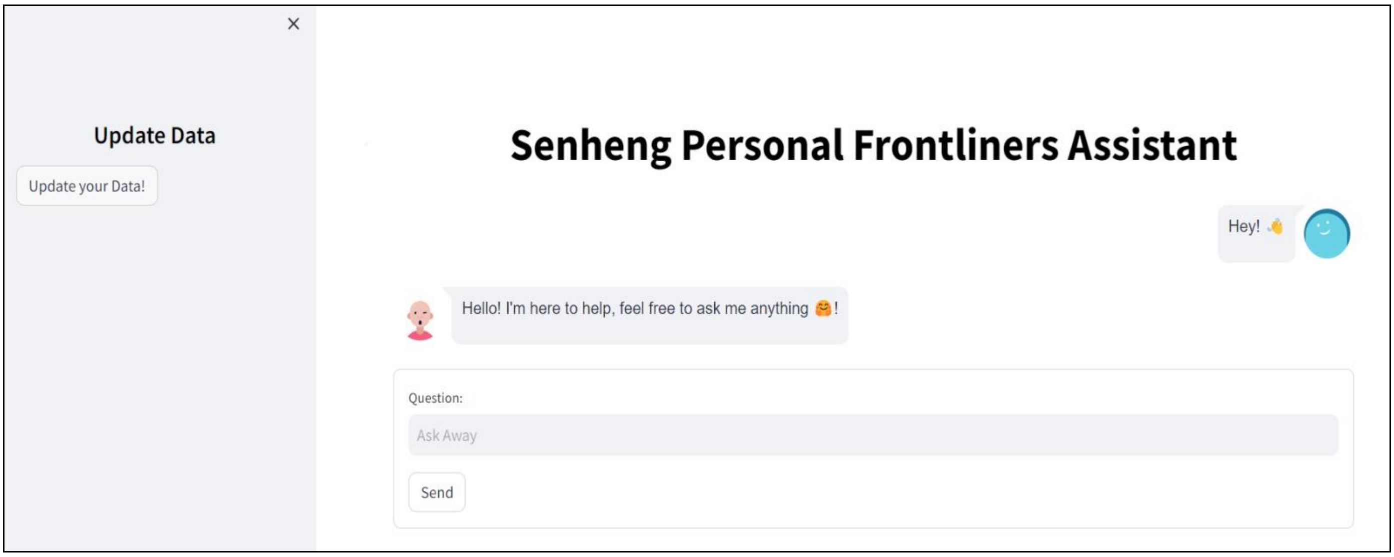
Lastly, the product_info column provides a concise description or review of the product. This column is the primary source of content used by the RAG model during the generation phase. It typically includes key features, performance metrics, and expert evaluations, providing rich, detailed context that the RAG model can draw upon to generate informative and accurate responses.

This structured data format enhances the RAG model's retrieval efficiency by allowing the retrieval component to function effectively, matching queries against product_name for relevance and using the corresponding product_info as context for generating responses. Additionally, the clear separation of product name and information ensures that the generation process is informed by comprehensive and specific data, leading to more accurate and contextually appropriate outputs. Furthermore, the organization of these columns supports scalability as the dataset grows, allowing new products to be added easily without disrupting the existing structure. This structured approach not only streamlines the operation of the chatbot but also significantly boosts its performance in delivering user-relevant content.

By adhering to these methodical and ethical data collection and preparation practices, we ensured that the RAG model could leverage high-quality, relevant data to enhance its performance. This foundation allows the chatbot to generate accurate and contextually appropriate responses based on up-to-date mobile phone reviews, significantly improving the user experience and reliability of the chatbot in real-world applications.

Refer to Appendix G on the code to setup Mistral 7B Model. For a Retrieval-Augmented Generation (RAG) model to function effectively, it is essential to format and structure the input data meticulously. In Appendix F, we discuss about the process of preparing the data to perform RAG.

B. *Overall Model System Development*



Senheng Personal Frontliners Assistant

Hey! 👋

Hello! I'm here to help, feel free to ask me anything 🤗!

Fig 9: Overall System View

Moving onto our frontliner chatbot assistant system, we have deployed it through Streamlit while implementing several features to enhance user experience and model performance. The outlook of the system is shown in Fig. 9.

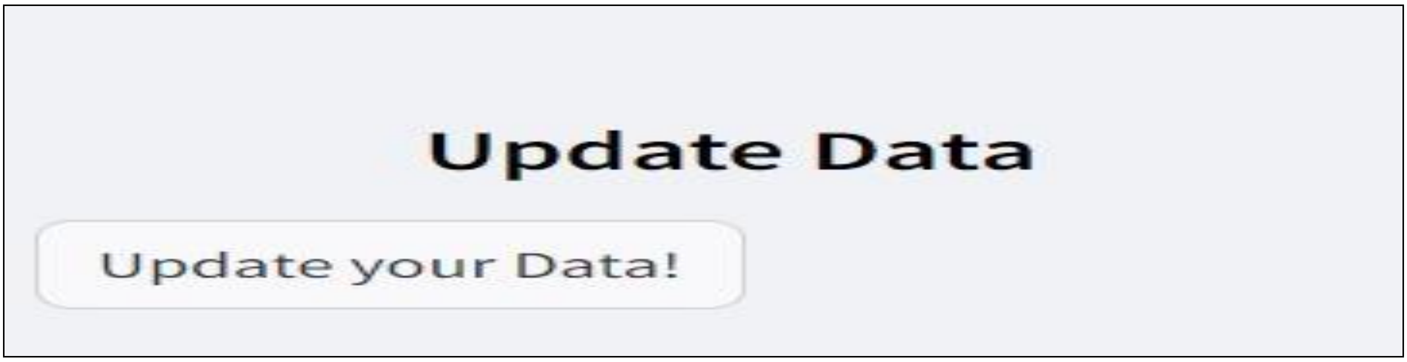
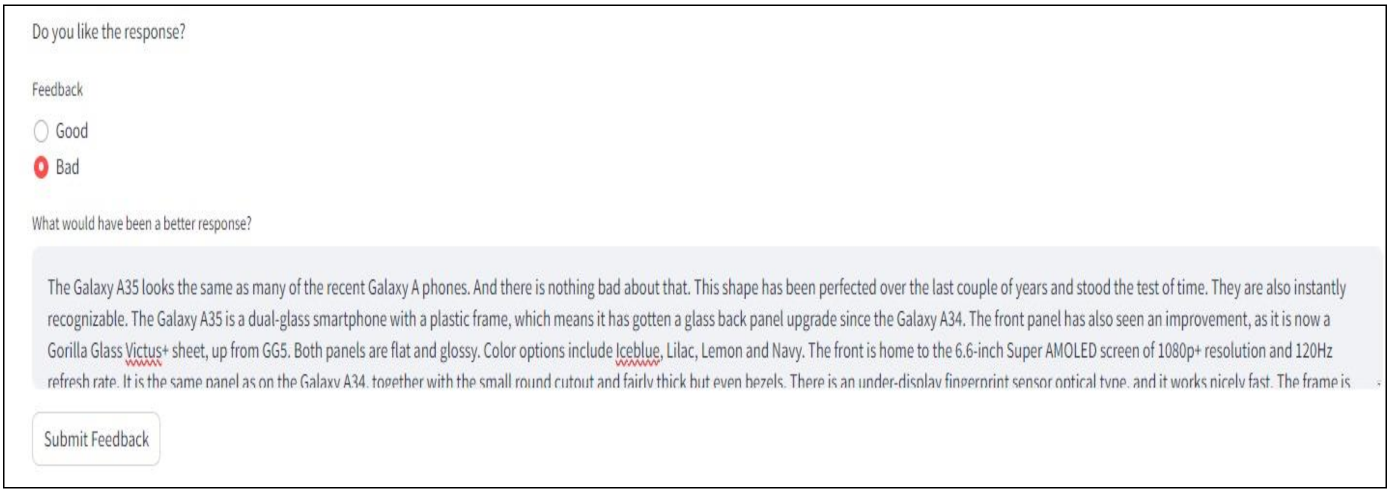


Fig 10: Update Dataset Button

By clicking on this button, the scraping script will then run in the background to update the dataset. This lets user to conveniently stay up to date with the chatbot according to Senheng's offerings on their website. The update data button is shown in Fig. 10.



Do you like the response?

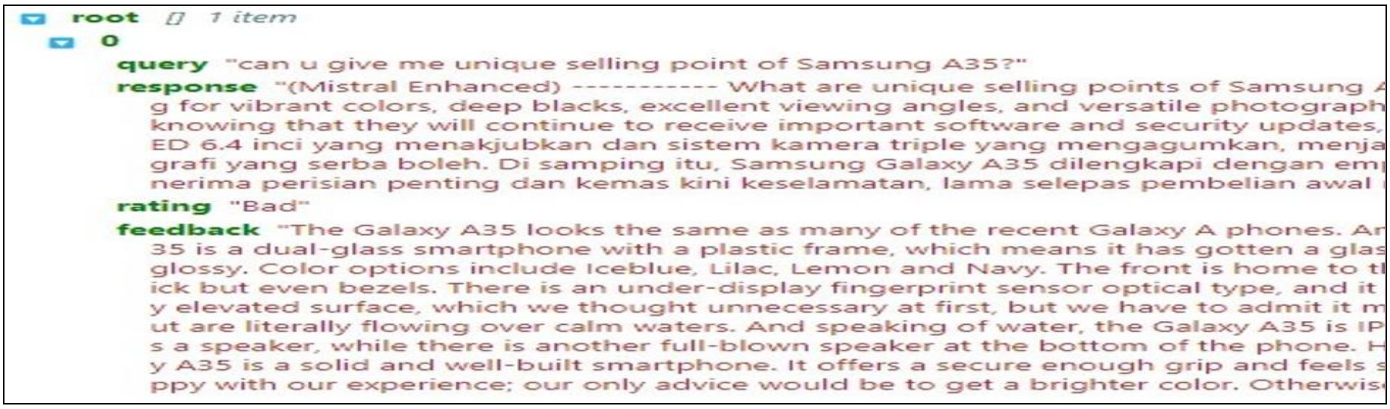
Feedback

○ Good
● Bad

What would have been a better response?

The Galaxy A35 looks the same as many of the recent Galaxy A phones. And there is nothing bad about that. This shape has been perfected over the last couple of years and stood the test of time. They are also instantly recognizable. The Galaxy A35 is a dual-glass smartphone with a plastic frame, which means it has gotten a glass back panel upgrade since the Galaxy A34. The front panel has also seen an improvement, as it is now a Gorilla Glass Victus+ sheet, up from GG5. Both panels are flat and glossy. Color options include Iceblue, Lilac, Lemon and Navy. The front is home to the 6.6-inch Super AMOLED screen of 1080p+ resolution and 120Hz refresh rate. It is the same panel as on the Galaxy A34, together with the small round cutout and fairly thick but even bezels. There is an under-display fingerprint sensor optical type, and it works nicely fast. The frame is

Fig 11: Feedback Function

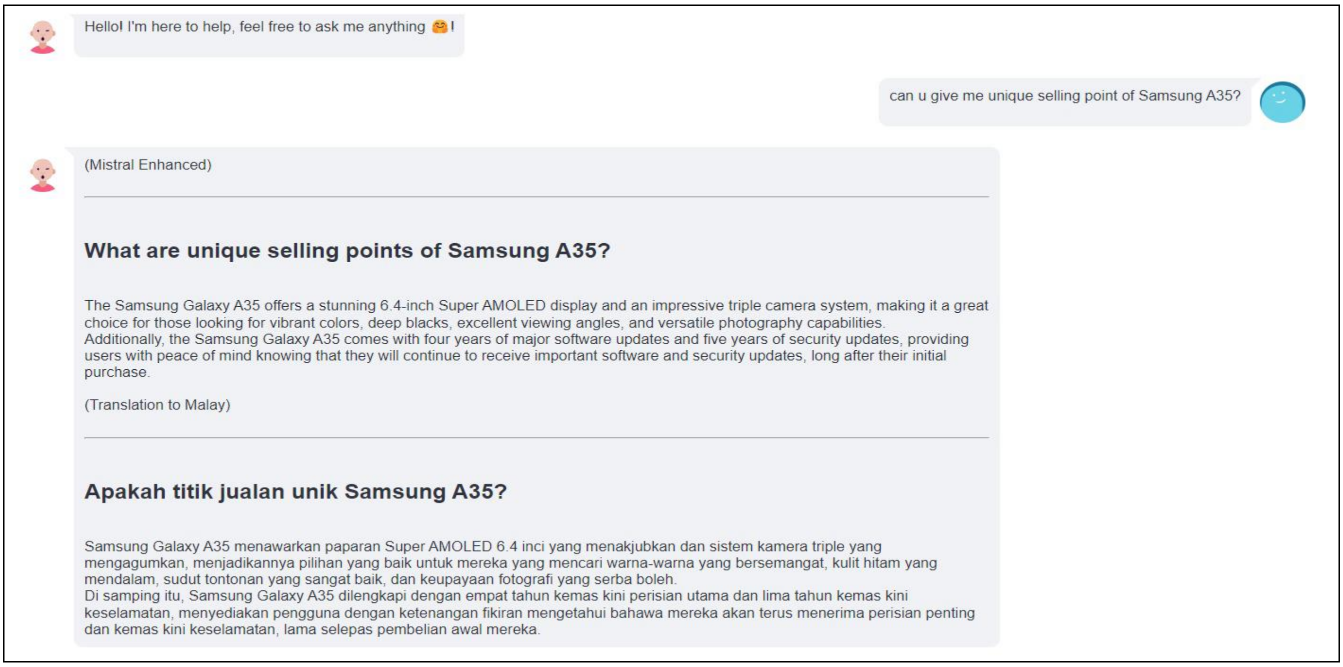


```
root  []  1 item
  0
    query  "can u give me unique selling point of Samsung A35?"
    response  "(Mistral Enhanced) ----------- What are unique selling points of Samsung A
       g for vibrant colors, deep blacks, excellent viewing angles, and versatile photograph
       knowing that they will continue to receive important software and security updates,
       ED 6.4 inci yang menakjubkan dan sistem kamera triple yang mengagumkan, menja
       grafi yang serba boleh. Di samping itu, Samsung Galaxy A35 dilengkapi dengan em
       nerima perisian penting dan kemas kini keselamatan, lama selepas pembelian awal
    rating  "Bad"
    feedback  "The Galaxy A35 looks the same as many of the recent Galaxy A phones. Ar
       35 is a dual-glass smartphone with a plastic frame, which means it has gotten a glas
       glossy. Color options include Iceblue, Lilac, Lemon and Navy. The front is home to t
       ick but even bezels. There is an under-display fingerprint sensor optical type, and it
       y elevated surface, which we thought unnecessary at first, but we have to admit it m
       ut are literally flowing over calm waters. And speaking of water, the Galaxy A35 is IP
       s a speaker, while there is another full-blown speaker at the bottom of the phone. H
       y A35 is a solid and well-built smartphone. It offers a secure enough grip and feels s
       ppy with our experience; our only advice would be to get a brighter color. Otherwis
```

Fig 12: Feedback Dataset

Besides that, we have included the feedback feature for fine tuning purposes to improve the model's relevancy and response generation in the future. User will be able to select "good" or "bad", submitting with the option "good" will then record the conversation and save it as pairs into the dataset. However, if "bad" is selected, a feedback box will appear to record the new preferred response by the user, and it will be saved subsequently into the json format dataset. Fig. 11 shows the feedback function and Fig. 12 shows an example of how the feedback dataset looks like.

Moving onto the chat function of the system, it can output dual language which are English and Bahasa Melayu to improve accessibility among users. Besides that, LLM Chaining is being implemented to refine the answers of the chatbot. In Appendix I, we discuss how LLM Chaining is being performed and used and the text is being translated in the program.



Hello! I'm here to help, feel free to ask me anything 🤗!

can u give me unique selling point of Samsung A35?

(Mistral Enhanced)

What are unique selling points of Samsung A35?

The Samsung Galaxy A35 offers a stunning 6.4-inch Super AMOLED display and an impressive triple camera system, making it a great choice for those looking for vibrant colors, deep blacks, excellent viewing angles, and versatile photography capabilities.
Additionally, the Samsung Galaxy A35 comes with four years of major software updates and five years of security updates, providing users with peace of mind knowing that they will continue to receive important software and security updates, long after their initial purchase.

(Translation to Malay)

Apakah titik jualan unik Samsung A35?

Samsung Galaxy A35 menawarkan paparan Super AMOLED 6.4 inci yang menakjubkan dan sistem kamera triple yang mengagumkan, menjadikannya pilihan yang baik untuk mereka yang mencari warna-warna yang bersemangat, kulit hitam yang mendalam, sudut tontonan yang sangat baik, dan keupayaan fotografi yang serba boleh.
Di samping itu, Samsung Galaxy A35 dilengkapi dengan empat tahun kemas kini perisian utama dan lima tahun kemas kini keselamatan, menyediakan pengguna dengan ketenangan fikiran mengetahui bahawa mereka akan terus menerima perisian penting dan kemas kini keselamatan, lama selepas pembelian awal mereka.

Fig 13: Chat log history

An example of a chat log from the chatbot was demonstrated in Fig. 13.

VI. DISCUSSION

The primary objective of this project was to enhance the overall productivity of product advisors in the consumer electronics retail industry. Through the successful implementation of a chatbot utilizing Mistral-7B and Retrieval-Augmented Generation (RAG), we are one step closer toward this goal. The chatbot is designed to interact with users and provide relevant, accurate responses to their queries, thereby increasing the efficiency and effectiveness of product advisors.

The integration of the Mistral-7B model with RAG has proven to be highly effective. The chatbot's ability to understand and respond to a wide range of inquiries allows product advisors to handle more customer interactions in less time. This not only boosts their productivity but also enhances the customer experience by providing timely and accurate information. The chatbot serves as an invaluable tool for junior product advisors, offering them guidance and support as they navigate customer inquiries. Its interactive capabilities ensure that even less experienced advisors can perform their duties with confidence, reducing their reliance on senior staff.

To ensure the chatbot remains up-to-date with the latest trends and products in the technology sector, we have incorporated an automatic update function. This feature enables the chatbot to crawl and retrieve relevant data from various sources, continually expanding its knowledge base. As a result, the chatbot can provide current and accurate information, keeping both the advisors and customers informed about new products and developments. This dynamic updating mechanism is crucial for maintaining the chatbot's relevance and effectiveness in a rapidly evolving industry.

VII. CONCLUSION AND FUTURE WORKS

While the current implementation of the chatbot has met our initial objectives, there is always room for further enhancement. Future iterations could include more sophisticated natural language processing (NLP) techniques to improve the chatbot's conversational abilities and context understanding. Additionally, integrating more advanced machine learning algorithms could enable the chatbot to offer predictive insights and personalized recommendations, further boosting its utility for both advisors and customers. Besides that, experts in this industry should be involved to vet the response generated by the chatbot to improve its accuracy and reliability.

In conclusion, the development and deployment of the chatbot using Mistral-7B and RAG have successfully addressed our primary objective of improving the productivity and efficiency of product advisors in the consumer electronics retail industry. The chatbot's ability to provide accurate, timely responses and its capacity for continuous learning will be invaluable to businesses in this ever-changing market. This project underscores the potential of AI-driven solutions like GenAI to transform traditional retail operations and enhance the capabilities of product advisors.

REFERENCES

[1]. Reinartz, W., Wiegand, N., & Imschloss, M. (2019). The Impact of Digital Transformation on the Retailing Value Chain. *International Journal of Research in Marketing*, *36*(3), 350–366. Science Direct. https://doi.org/10.1016/j.ijresmar.2018.12.002

[2]. AIContentfy (2023) *The role of customer education in customer acquisition and retention*. (2023, February 11). AIContentfy. https://aicontentfy.com/en/blog/role-of-customer-education-in-customer-acquisition-and-retention

[3]. Kim, J.-H., Kim, M., Park, M., & Yoo, J. (2022). Immersive Interactive Technologies and Virtual Shopping Experiences: Differences in Consumer Perceptions between Augmented Reality (AR) and Virtual Reality (VR). *Telematics and Informatics*, *77*, 101936. https://doi.org/10.1016/j.tele.2022.101936

[4]. Hopkins, M. (2023, February 21). *Building a competitive advantage in retail with technology*. Planning and Analytics Blog – Board International. https://blog.board.com/building-competitive-advantage-retail-technology-is-key/

[5]. Varnika Om (2022) *Customer Service is a Key Differentiator in Retail*. (2022, May 27). Freshdesk Blogs. https://www.freshworks.com/freshdesk/customer-support/customer-service-in-retail-a-differentiator-blog/

[6]. Netstock. (2023, June 13). *Demand Forecasting for Supply Chains: How to Predict & Plan*. Netstock. https://www.netstock.com/blog/demand-forecasting-for-supply-chains-how-to-predict-plan/

[7]. Mahmoud, M. A., Tsetse, E. K. K., Tulasi, E. E., & Muddey, D. K. (2022). Green Packaging, Environmental Awareness, Willingness to Pay and Consumers' Purchase Decisions. Sustainability, 14(23), 16091. https://doi.org/10.3390/su142316091

[8]. PROS, Inc. (2023, March 29). *Competitive Pricing Strategy: Benefits and Disadvantages*. Pros.com. https://pros.com/learn/b2b-blog/competitive-pricing-strategy

[9]. Anchanto. (2022, May 1). 6 Struggles E-commerce Businesses in Consumer Electronics Know Too Well + (Proven Solutions). Anchanto. https://anchanto.com/6-struggles-ecommerce-businesses-in-consumer-electronics-know-too-well-proven-solutions/

[10]. Chong, T., Yu, T., Keeling, D. I., & de Ruyter, K. (2021). AI-chatbots on the services frontline addressing the challenges and opportunities of agency. *Journal of Retailing and Consumer Services*, *63*, 102735. https://doi.org/10.1016/j.jretconser.2021.102735

[11]. Pantano, E., & Pizzi, G. (2020). Forecasting artificial intelligence on online customer assistance: Evidence from chatbot patents analysis. *Journal of Retailing and Consumer Services*, *55*, 102096. https://doi.org/10.1016/j.jretconser.2020.102096

[12]. Shankar, V. (2018). How Artificial Intelligence (AI) is Reshaping Retailing. *Journal of Retailing*, *94*(4), vi–xi. https://doi.org/10.1016/s0022-4359(18)30076-9

[13]. Bhattacharyya, R., Chandra, S., Manikonda, K., Depuru, B., & Kumar, B. (2024). An AI-Driven Interactive Chatbot: A Well-Trained Chatbot that Communicates with the Users and Reduces the Manual Interaction. *International Journal of Innovative Science and Research Technology*, *9*(2)

[14]. Guha, A., Grewal, D., Kopalle, P. K., Haenlein, M., Schneider, M. J., Jung, H., Moustafa, R., Hegde, D. R., & Hawkins, G. (2021). How artificial intelligence will affect the future of retailing. *Journal of Retailing*, *97*(1), 28–41. https://doi.org/10.1016/j.jretai.2021.01.005

[15]. Kaur, V., Khullar, V., & Verma, N. (2020). Review of Artificial Intelligence with retailing sector. *Journal of Computer Science Research*, *2*(1). https://doi.org/10.30564/jcsr.v2i1.1591

[16]. Kim, J., & Min, M. (2024). From RAG to QA-RAG: Integrating Generative AI for Pharmaceutical Regulatory Compliance Process.

[17]. Moore, S., Bulmer, S., & Elms, J. (2022). The social significance of AI in retail on customer experience and shopping practices. *Journal of Retailing and Consumer Services*, *64*(1), 102755. https://doi.org/10.1016/j.jretconser.2021.102755

[18]. Seranmadevi, R., & Senthil Kumar, A. (2019). Experiencing the AI emergence in Indian retail – Early adopters approach. *Management Science Letters*, 33–42. https://doi.org/10.5267/j.msl.2018.11.002

[19]. Trivedi, S., & Patel, N. (2020). The Role of Automation and Artificial Intelligence in Increasing the Sales Volume: Evidence from M, S, and, MM Regressions. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4180379

[20]. Tuan, N. T., Moore, P., Thanh, D. H. V., & Pham, H. V. (2024). A Generative Artificial Intelligence Using Multilingual Large Language Models for ChatGPT Applications. *Applied Sciences*, *14*(7), 3036. https://doi.org/10.3390/app14073036

[21]. Dhake, S. (2024). Impacts and Implications of Generative AI and Large Language Models: Redefining Banking Sector. *Impacts and Implications of Generative AI and Large Language Models: Redefining Banking Sector*, *14*(2).

[22]. Dhoni, P. S. (2024). From Data to Decisions: Enhancing Retail with AI and Machine Learning. *International Journal of Computing and Engineering*, *5*(1), 38–51. https://doi.org/10.47941/ijce.1660

[23]. Ding, M., Dong, S., & Grewal, R. (2024). Generative AI and Usage in Marketing Classroom. *Customer Needs and Solutions*, *11*(1). https://doi.org/10.1007/s40547-024-00145-2

[24]. Gao, Y., Arava, S. K., Li, Y., & Jr, J. Ws. (2024). Improving the Capabilities of Large Language Model based Marketing Analytics Copilots with Semantic Search and Fine-Tuning. *International Journal on Cybernetics & Informatics*, *13*(2), 15–31. https://doi.org/10.5121/ijci.2024.130202

[25]. Expert Marketing Advisors. (2023, September 14). *Unlocking A Seamless Customer Experience With Omnichannel Marketing*. Www.linkedin.com. https://www.linkedin.com/pulse/unlocking-seamless-customer-experience-omnichannel

[26]. Cook, B. (2023, August 22). *Cost Reduction Process: Definition and Steps*. Tipalti.com. https://tipalti.com/expenses-hub/cost-reduction-process/

[27]. Cohen, S. (2022, September 19). *The Growing Awareness and Prominence of Environmental Sustainability*. State of the Planet. https://news.climate.columbia.edu/2022/09/19/the-growing-awareness-and-prominence-of-environmental-sustainability/

[28]. Reddy, K. Pradeep. (2023). Consumers perception on green marketing towards eco-friendly fast moving consumer goods. *International Journal of Engineering Business Management*, *15*(15). https://doi.org/10.1177/18479790231170962

[29]. Acxiom Technologies LLP. (2023, October 30). *Advantages Of Using AI chatbots to Strengthen IT Service Management*. Www.linkedin.com. https://www.linkedin.com/pulse/advantages-using-ai-chatbots-strengthen-service-management-qwr1f

[30]. LaMorte, W. (2022). *The Social Cognitive Theory*. Boston University School of Public Health. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/SB/BehavioralChangeTheories/BehavioralChangeTheories5.html

[31]. Ostojić, I. (2024, May 14). *Council Post: More Than Chatbots: AI Trends Driving Conversational Experiences For Customers*. Forbes. https://www.forbes.com/sites/forbesbusinesscouncil/2024/03/15/more-than-chatbots-ai-trends-driving-conversational-experiences-for-customers/?sh=23b772bb29f7

[32]. Mehta, J. (2023, November 18). *The role of chatbots in personalized customer experiences*. Abmatic.ai. https://abmatic.ai/blog/role-of-chatbots-in-personalized-customer-experiences#:~:text=Personalized%20interactions%3A%20Chatbots%20can%20use

[33]. Trend Hunter. (2023, April 28). *How AI is Revolutionizing Retail*. Www.linkedin.com. https://www.linkedin.com/pulse/how-ai-revolutionizing-retail-trend-hunter

[34]. Fergus, S. (2024, January 17). *Exponential Growth of Data: Avoiding a Resource Allocation Paradox*. Shipyard. https://www.shipyardapp.com/blog/exponential-data-growth/#:~:text=As%20data%20volumes%20grow%20exponentially

[35]. Stalmachova, K., Chinoracky, R., & Strenitzerova, M. (2022). Changes in Business Models Caused by Digital Transformation and the COVID-19 Pandemic and Possibilities of Their Measurement—Case Study. *Sustainability*, *14*(1), 127. Mdpi. https://doi.org/10.3390/su14010127

[36]. Buehler, T. Leigh. (2024, April 3). *Artificial Intelligence in Retail and Improving Efficiency | American Public University*. Www.apu.apus.edu. https://www.apu.apus.edu/area-of-study/business-and-management/resources/artificial-intelligence-in-retail-and-improving-efficiency/

[37]. Coherent Market Insights. (2023, December 26). *The Future of Shopping: How AI is Reshaping Retail Dynamics*. Www.linkedin.com. https://www.linkedin.com/pulse/future-shopping-how-ai-reshaping-retail-dynamics-mkijf

[38]. Grewal, D., Benoit, S., Noble, S., Guha, A., Carl-Philip Ahlbom, & Jens Nordfält. (2023). Leveraging In-Store Technology and AI: Increasing Customer and Employee Efficiency and Enhancing their Experiences. *Journal of Retailing*, *99*(4). https://doi.org/10.1016/j.jretai.2023.10.002

[39]. Ruparelia, A. (2022, March 5). *8 Types of Traditional Marketing Methods to Implement - Salespanel*. Salespanel Blog. https://salespanel.io/blog/marketing/traditional-marketing-methods/

[40]. Hashemi-Pour, C. (2023, October). *What Is CRM (Customer Relationship Management)?* TechTarget. https://www.techtarget.com/searchcustomerexperience/definition/CRM-customer-relationship-management

[41]. Susanti, Y., Pratiwi, H., Sri Sulistijowati, H., & Liana, T. (2014). P A. *M Estimation, S Estimation, and MM Estimation in Robust Regression*, *Volume 91*(No. 3). https://doi.org/10.12732/ijpam.v91i3.7

[42]. Dwivedi, Y. K., Ismagilova, E., Hughes, D. L., & Carlson, J. (2021). Setting the Future of Digital and Social Media Marketing research: Perspectives and Research Propositions. *International Journal of Information Management*, *59*(1), 1–37. Sciencedirect. https://doi.org/10.1016/j.ijinfomgt.2020.102168

[43]. Integration, T. (2023, August 16). *5 ways to streamline & improve your supply chain process - 2023*. Target Integration. https://targetintegration.com/5-ways-to-streamline-improve-your-supply-chain-process-2023

[44]. V-Count. (2024, January 26). *The Impact of AI Technologies in Retail - V-Count*. V-Count.com. https://v-count.com/the-impact-of-ai-technologies-on-retail-examining-benefits-and-transformations-in-shopping/#:~:text=AI%2C%20with%20its%20advanced%20algorithms

[45]. Ngoc, B. (2024, April 21). *Why Affordable AI Chatbots Can Revolutionize Your SME*. Www.linkedin.com.

[46]. https://www.linkedin.com/pulse/why-affordable-ai-chatbots-can-revolutionize-your-sme-bui-ngoc-okr1c?trk=public_post

[47]. Fernandex, R. (2024, May 5). *Mistral AI Launches Mistral Large LLM and "Le Chat": Everything You Need to Know*. Techopedia. https://www.techopedia.com/mistral-ai-launches-mistral-large-llm-and-le-chat#:~:text=Mistral%20Large's%20capacities%20are%20still,as%20a%20text%2Dbased%20model.

[48]. LangChain. (2024). *Q&A with RAG | LangChain*. Python.langchain.com. https://python.langchain.com/v0.1/docs/use_cases/question_answering/

[49]. MoldStud. (2024, May 5). *Benefits of using AI-driven chatbots in software applications*. Moldstud.com. https://moldstud.com/articles/p-benefits-of-using-ai-driven-chatbots-in-software-applications

[50]. Ali, O., Murray, P. A., Momin, M., Dwivedi, Y. K., & F. Tegwen Malik. (2024). The effects of artificial intelligence applications in educational settings: Challenges and strategies. *Technological Forecasting and Social Change*, *199*, 123076–123076. https://doi.org/10.1016/j.techfore.2023.123076

[51]. Hackett, T. (2024, February 23). *Competitive Pricing Tactics: Stay Profitable in a Cutthroat Market*. Medium. https://medium.com/@tony.hackett/competitive-pricing-tactics-stay-profitable-in-a-cutthroat-market-d3d0f4f43926

[52]. Walia, E. (2024, January 11). *Cybersecurity essentials for Retail business in 2024*. Www.linkedin.com. https://www.linkedin.com/pulse/cybersecurity-essentials-retail-business-2024-eshan-walia-wy3ef

[53]. Božić, V. (2023). RISKS OF DIGITAL DIVIDE IN USING ARTIFICAL INTELLIGENCE (AI). *RISKS of DIGITAL DIVIDE in USING ARTIFICAL INTELLIGENCE (AI)*. https://doi.org/10.13140/RG.2.2.18156.13443

[54]. Elsaid, H. (2024, March 1). *AI in banking: What will it actually change?* Trade Finance Global. https://www.tradefinanceglobal.com/posts/ai-in-banking-what-will-it-actually-change/

[55]. Hypersonix, & Becchetti, G. (2024, March 28). *Optimizing Retail Operations with Predictive Analytics and AI*. Hypersonix. https://hypersonix.ai/blog/optimizing-retail-operations-with-predictive-analytics-and-ai/

[56]. Joadekar, P. (2024). *AI and Responsible Banking: Balancing Efficiency with Ethics | Synechron*. Www.synechron.com. https://www.synechron.com/insight/ai-and-responsible-banking-balancing-efficiency-ethics

[57]. OnFinance AI. (2024, April 14). *Unlocking Precision: The Art of Fine-Tuning Language Models*. Www.linkedin.com. https://www.linkedin.com/pulse/unlocking-precision-art-fine-tuning-language-models-fwwfc

[58]. Pandey, P. (2023, August 10). *Exploring Semantic Search Using Embeddings and Vector Databases with some popular Use Cases*. Medium. https://medium.com/@pankaj_pandey/exploring-semantic-search-using-embeddings-and-vector-databases-with-some-popular-use-cases-2543a79d3ba6

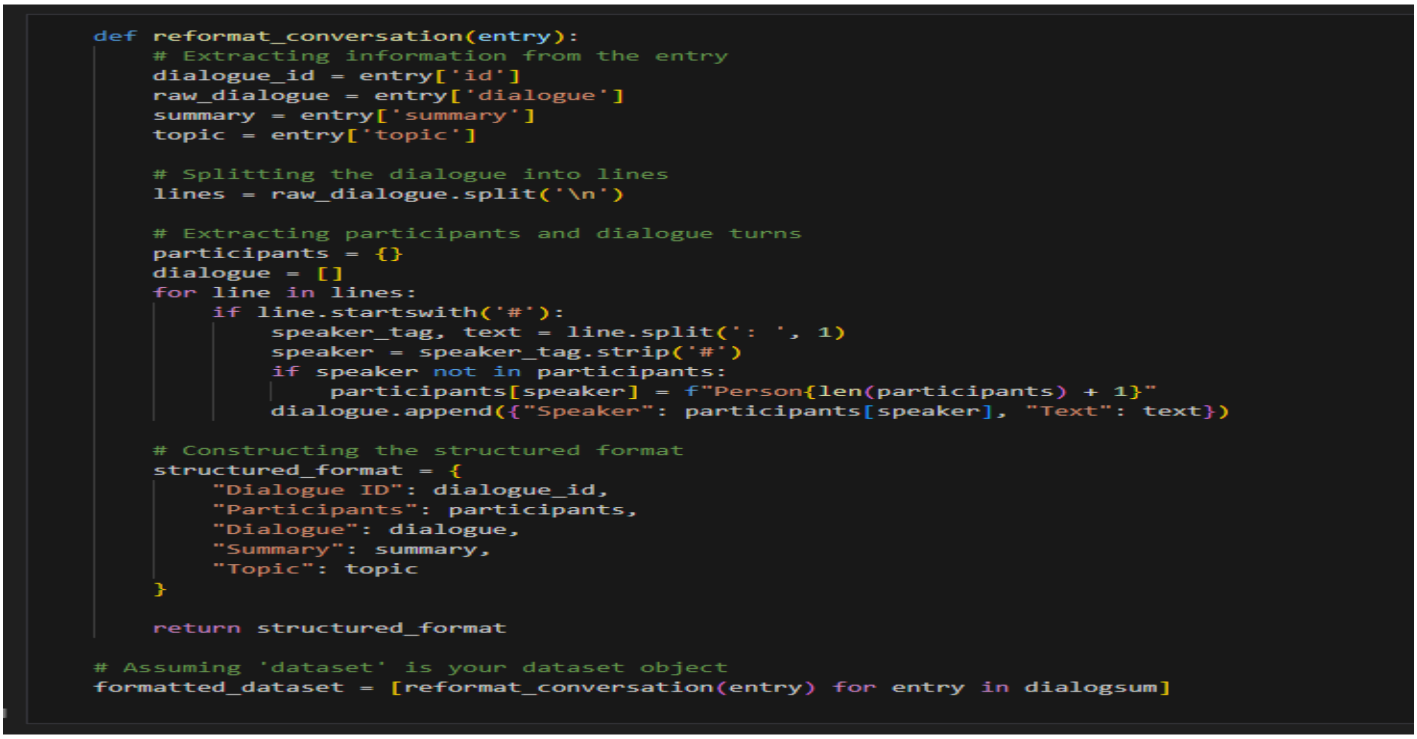
[59]. Rizvi, M. (2023). Investigating AI-Powered Tutoring Systems that Adapt to Individual Student Needs, Providing Personalized Guidance and Assessments. *The Eurasia Proceedings of Educational and Social Sciences*, *31*, 67–73. https://doi.org/10.55549/epess.1381518

[60]. Roy, M. (2022). Artificial Intelligence in Pharmaceutical Sales & Marketing -A Conceptual Overview.

[61]. Oosthuizen, K., Botha, E., Robertson, J., & Montecchi, M. (2020). Artificial intelligence in retail: The AI-enabled value chain. *Australasian Marketing Journal (AMJ)*, *29*(3). https://doi.org/10.1016/j.ausmj.2020.07.007

APPENDICES

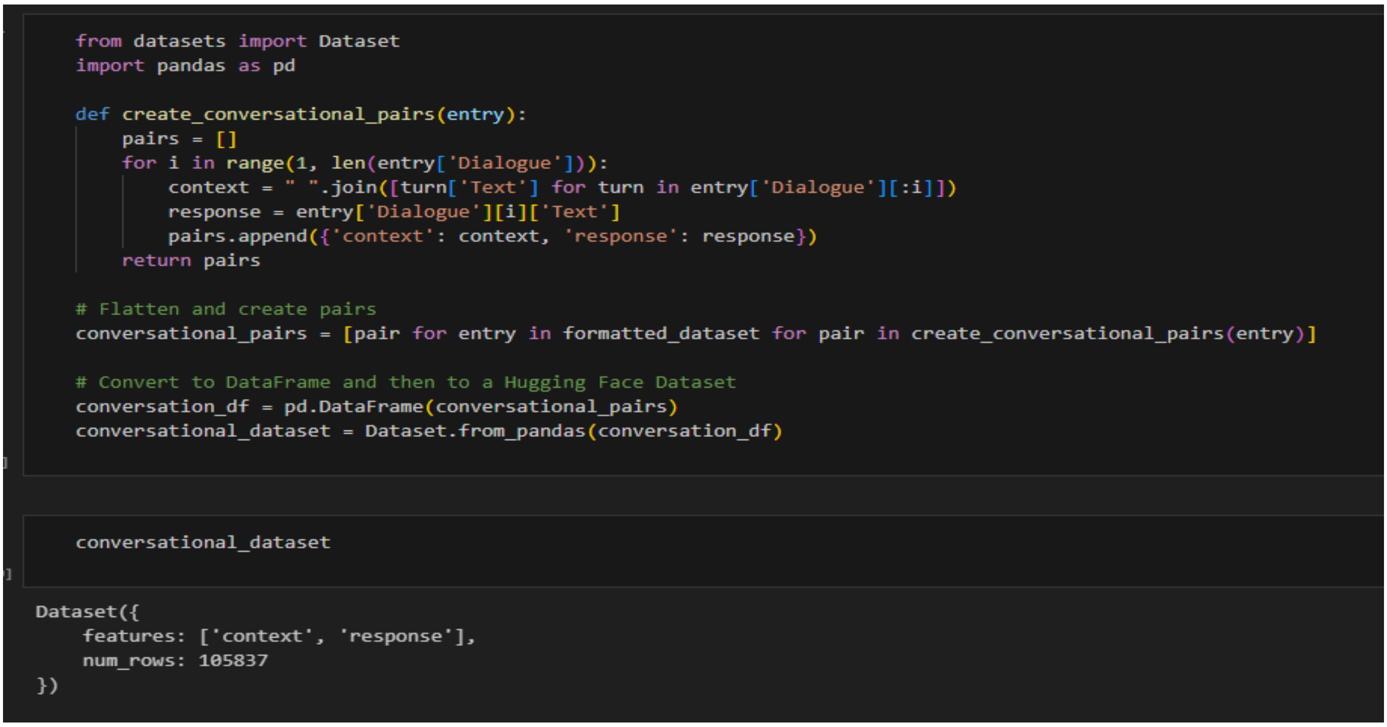
A. Data Preparation Code



```
def reformat_conversation(entry):
    # Extracting information from the entry
    dialogue_id = entry['id']
    raw_dialogue = entry['dialogue']
    summary = entry['summary']
    topic = entry['topic']

    # Splitting the dialogue into lines
    lines = raw_dialogue.split('\n')

    # Extracting participants and dialogue turns
    participants = {}
    dialogue = []
    for line in lines:
        if line.startswith('#'):
            speaker_tag, text = line.split(': ', 1)
            speaker = speaker_tag.strip('#')
            if speaker not in participants:
                participants[speaker] = f"Person{len(participants) + 1}"
            dialogue.append({"Speaker": participants[speaker], "Text": text})

    # Constructing the structured format
    structured_format = {
        "Dialogue ID": dialogue_id,
        "Participants": participants,
        "Dialogue": dialogue,
        "Summary": summary,
        "Topic": topic
    }

    return structured_format

# Assuming 'dataset' is your dataset object
formatted_dataset = [reformat_conversation(entry) for entry in dialogsum]
```

Fig 14: Code for reformat_conversation

The dataset is first being reformatted so that it will be easier to preprocess the dataset and create conversational pairs which is in a format that we need to fine-tune the Falcon7B model further down the line shown in Fig. 14.



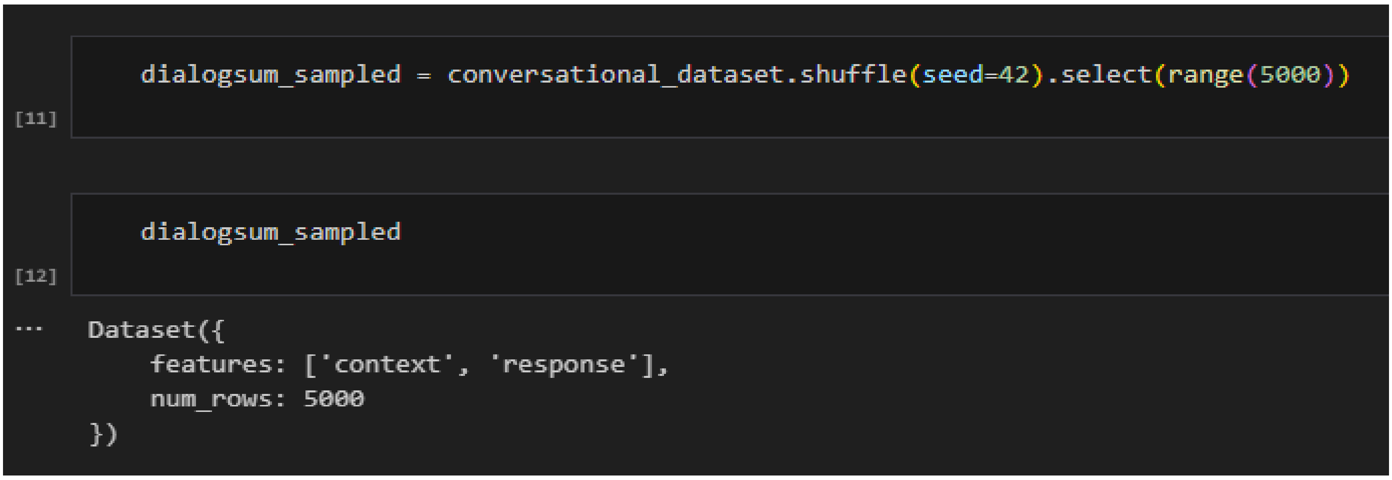
```
from datasets import Dataset
import pandas as pd

def create_conversational_pairs(entry):
    pairs = []
    for i in range(1, len(entry['Dialogue'])):
        context = " ".join([turn['Text'] for turn in entry['Dialogue'][:i]])
        response = entry['Dialogue'][i]['Text']
        pairs.append({'context': context, 'response': response})
    return pairs

# Flatten and create pairs
conversational_pairs = [pair for entry in formatted_dataset for pair in create_conversational_pairs(entry)]

# Convert to DataFrame and then to a Hugging Face Dataset
conversation_df = pd.DataFrame(conversational_pairs)
conversational_dataset = Dataset.from_pandas(conversation_df)
```

```
conversational_dataset
```

```
Dataset({
    features: ['context', 'response'],
    num_rows: 105837
})
```

Fig 15: Code to Form Conversation Pairs

The code above in Fig. 15 shows that we have successfully preprocessed the dataset and pre-processed it to a suitable format for fine-tuning. The model Falcon7B requires a dataset that contains only columns “context” and “response” to be fine-tuned.

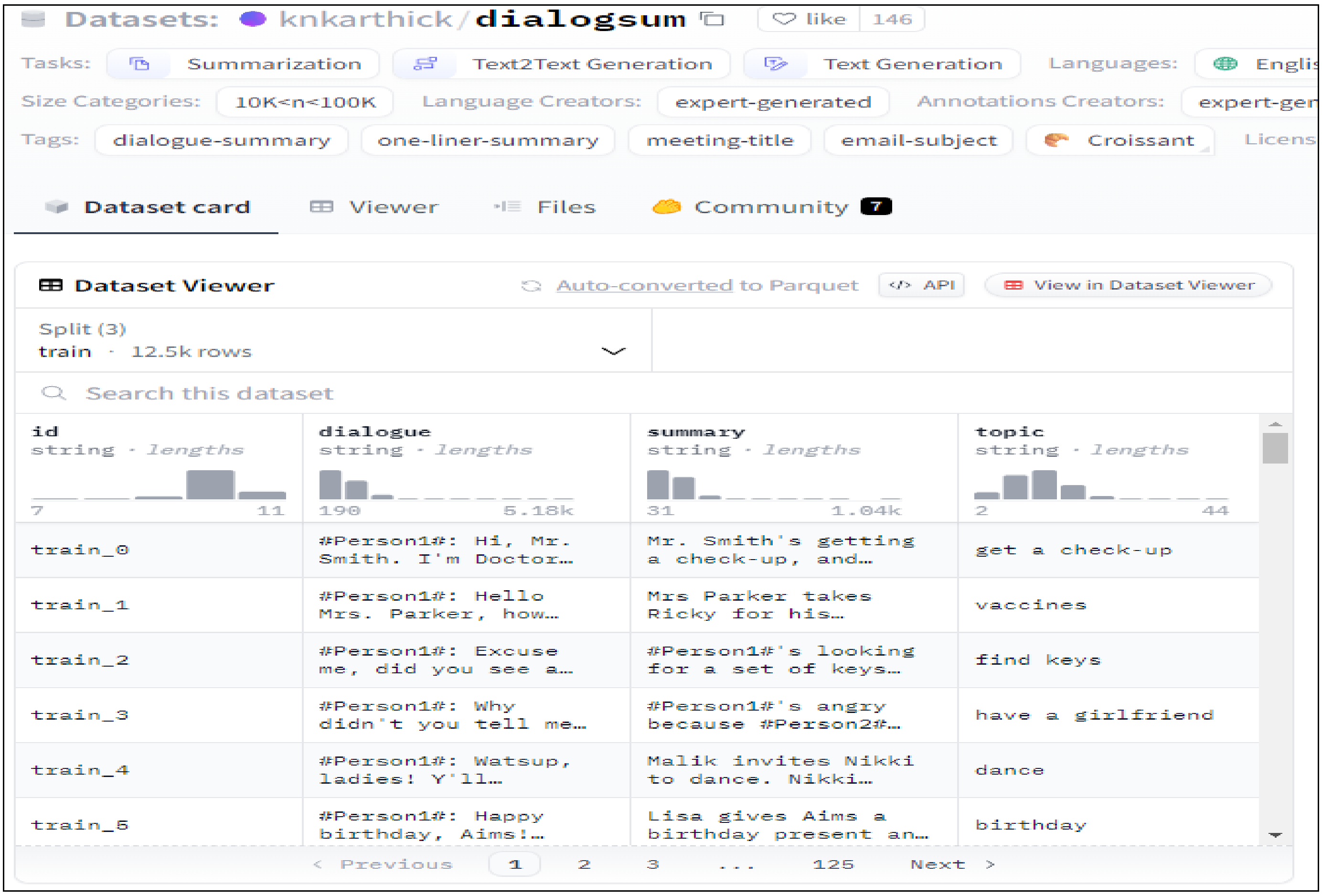


```
[11] dialogsum_sampled = conversational_dataset.shuffle(seed=42).select(range(5000))

[12] dialogsum_sampled

...  Dataset({
         features: ['context', 'response'],
         num_rows: 5000
     })
```

Fig 16: Code to Select 5k Random Rows

However, due to computational resources and time constraints, we will not be able to fine-tune the model with the entire dataset as it will take a very long time to complete the fine-tuning process. Hence 5 thousand rows have been randomly selected as shown in Fig. 16 to test whether the model can properly generate responses.

B. Data Preparation for Falcon7B

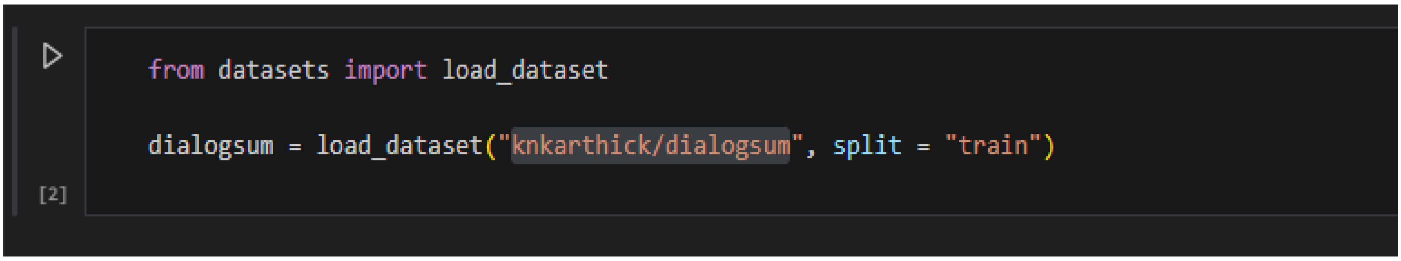


Datasets: knkarthick/dialogsum
like 146
Tasks: Summarization Text2Text Generation Text Generation Languages: Englis
Size Categories: 10K<n<100K Language Creators: expert-generated Annotations Creators: expert-ger
Tags: dialogue-summary one-liner-summary meeting-title email-subject Croissant Licens
Dataset card Viewer Files Community 7
Dataset Viewer Auto-converted to Parquet API View in Dataset Viewer
Split (3)
train · 12.5k rows
Search this dataset

id string · lengths 7 – 11	dialogue string · lengths 190 – 5.18k	summary string · lengths 31 – 1.04k	topic string · lengths 2 – 44
train_0	#Person1#: Hi, Mr. Smith. I'm Doctor…	Mr. Smith's getting a check-up, and…	get a check-up
train_1	#Person1#: Hello Mrs. Parker, how…	Mrs Parker takes Ricky for his…	vaccines
train_2	#Person1#: Excuse me, did you see a…	#Person1#'s looking for a set of keys…	find keys
train_3	#Person1#: Why didn't you tell me…	#Person1#'s angry because #Person2#…	have a girlfriend
train_4	#Person1#: Watsup, ladies! Y'll…	Malik invites Nikki to dance. Nikki…	dance
train_5	#Person1#: Happy birthday, Aims!…	Lisa gives Aims a birthday present an…	birthday

Fig 17: Data Source from Hugging Face

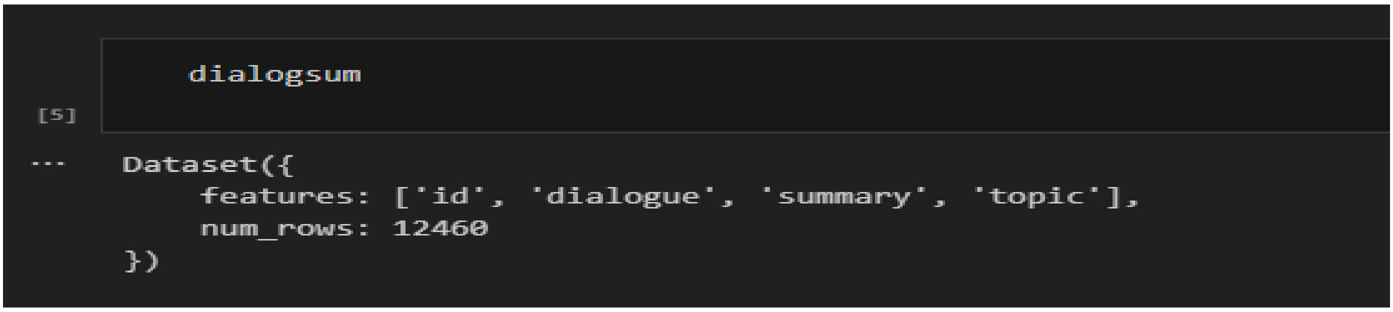
To start, a suitable dataset is found from Hugging Face to fine-tune our model Falcon7B. The dataset that will be used in this fine-tuning process would be the dialogsum dataset provided by knkarthick on Hugging Face in Fig. 17.



```
from datasets import load_dataset

dialogsum = load_dataset("knkarthick/dialogsum", split = "train")
```

Fig 18: Downloading Dataset using “Datasets” Library

The dataset is then downloaded using the datasets library using the code in Fig. 18.

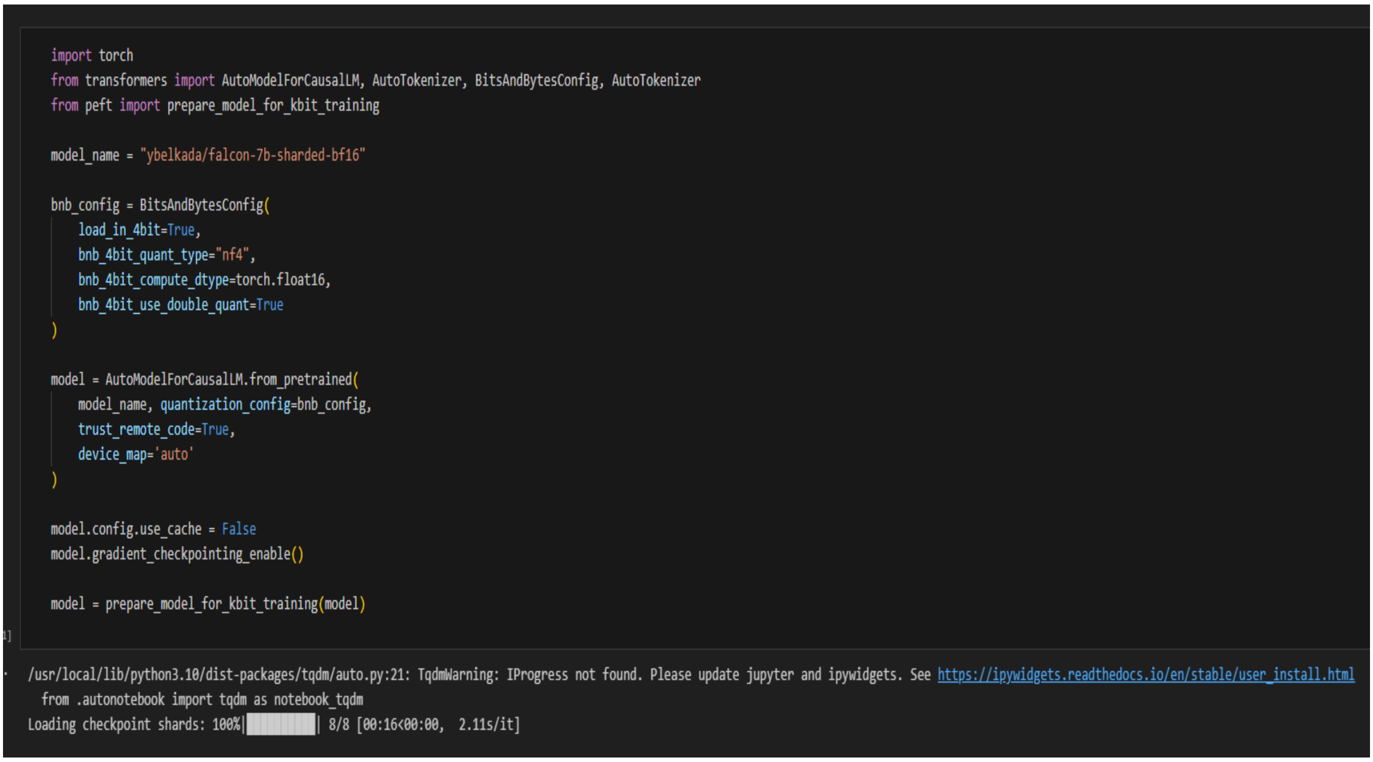


```
dialogsum
```

```
Dataset({
    features: ['id', 'dialogue', 'summary', 'topic'],
    num_rows: 12460
})
```

Fig 19: Detailed Structure of Dataset

From the screenshot above in Fig. 19, we can observe the original structure of the dataset.



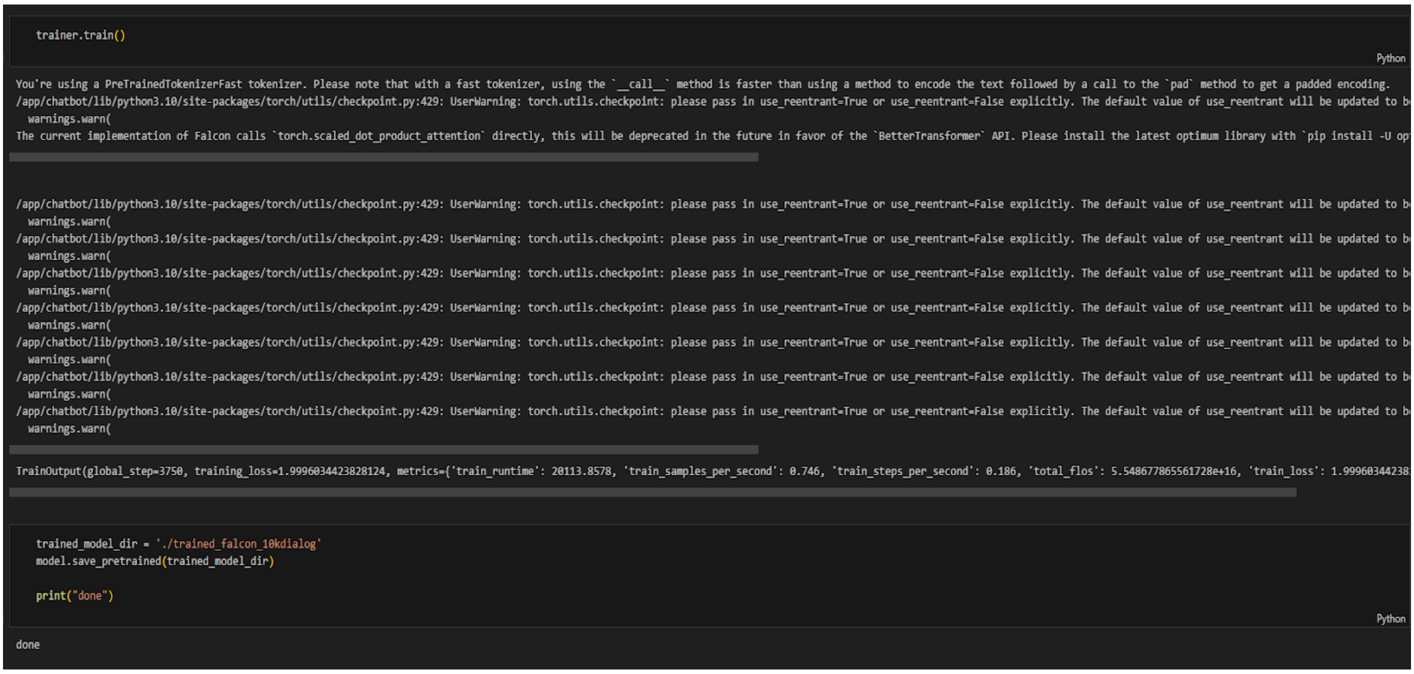
```
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig, AutoTokenizer
from peft import prepare_model_for_kbit_training

model_name = "ybelkada/falcon-7b-sharded-bf16"

bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
    bnb_4bit_use_double_quant=True
)

model = AutoModelForCausalLM.from_pretrained(
    model_name, quantization_config=bnb_config,
    trust_remote_code=True,
    device_map='auto'
)

model.config.use_cache = False
model.gradient_checkpointing_enable()

model = prepare_model_for_kbit_training(model)
```

```
/usr/local/lib/python3.10/dist-packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
Loading checkpoint shards: 100%|██████████| 8/8 [00:16<00:00,  2.11s/it]
```

Fig 20: Importing Model

From the screenshot above in Fig. 20, the model is being imported.

C. Falcon7B Inferencing and RAG



```
trainer.train()
```

Python

```
You're using a PreTrainedTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the text followed by a call to the `pad` method to get a padded encoding.
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
The current implementation of Falcon calls `torch.scaled_dot_product_attention` directly, this will be deprecated in the future in favor of the `BetterTransformer` API. Please install the latest optimum library with `pip install -U op


/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(
/app/chatbot/lib/python3.10/site-packages/torch/utils/checkpoint.py:429: UserWarning: torch.utils.checkpoint: please pass in use_reentrant=True or use_reentrant=False explicitly. The default value of use_reentrant will be updated to b
  warnings.warn(

TrainOutput(global_step=3750, training_loss=1.9996034423828124, metrics={'train_runtime': 20113.8578, 'train_samples_per_second': 0.746, 'train_steps_per_second': 0.186, 'total_flos': 5.548677865561728e+16, 'train_loss': 1.99960344238
```

```
trained_model_dir = './trained_falcon_10kdialog'
model.save_pretrained(trained_model_dir)

print("done")
```

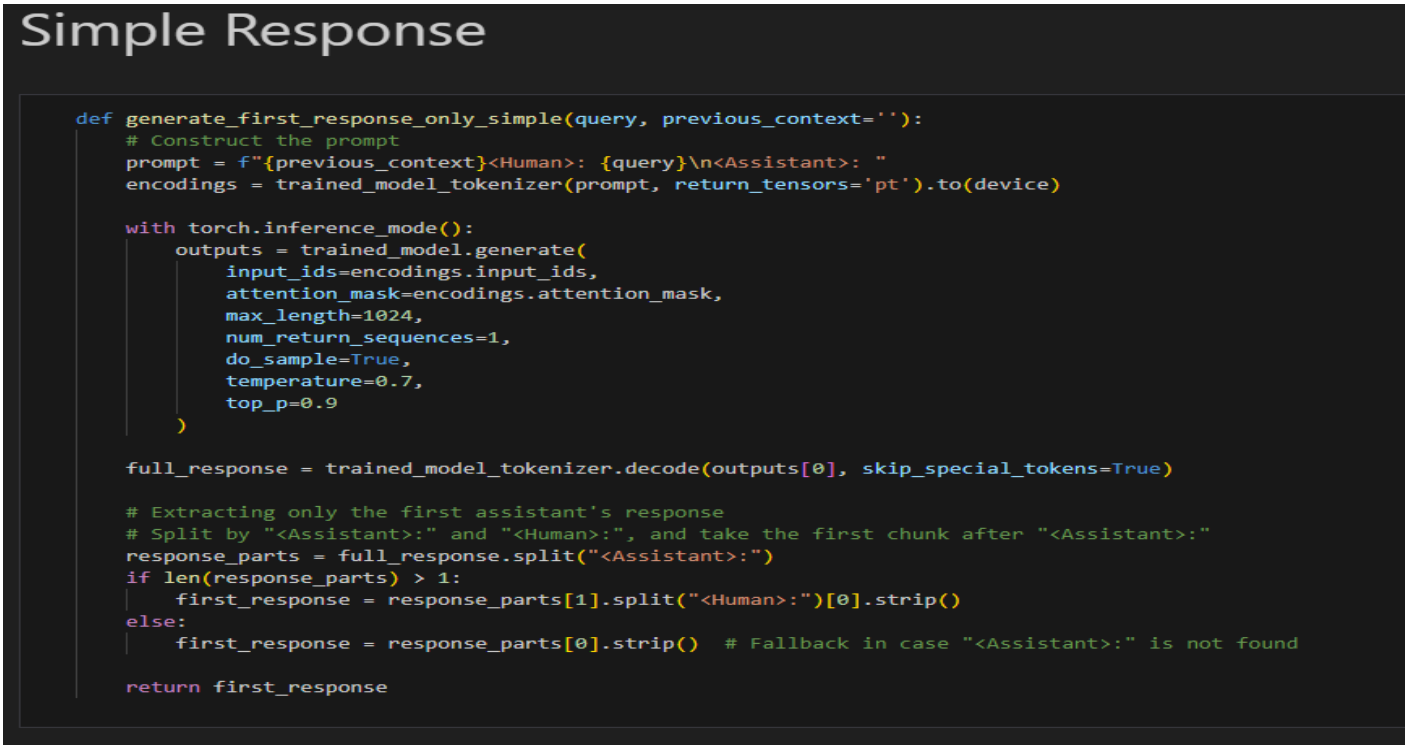
Python

```
done
```

Fig 21: Initializing Tokenization

Next, as shown in Fig. 21, we downloaded the model and save the checkpoints using the HuggingFace API. In this case, we have used the sharded version of the original Falcon7B because we want to finetune it.

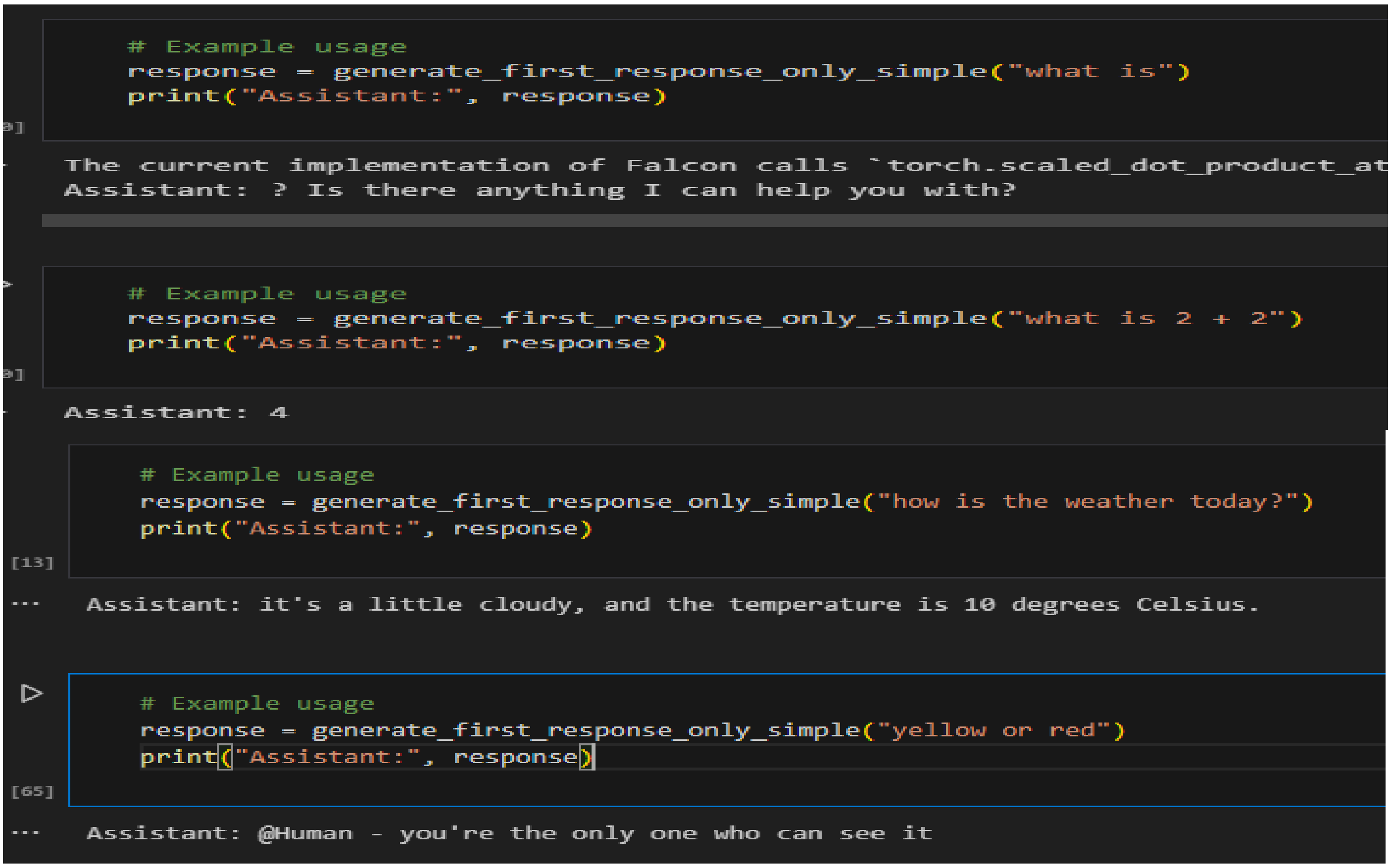
After setting up the model (tokenizations, peft configs etc), we can now start to finetune the model using the train function. This process took about 5.5 hours with the epoch being set to 3. This resulted in a training loss of 1.999603442382812. After training, the model is then saved into a folder with files like adapter_config.json and adapter_model.bin present.



Simple Response

```
def generate_first_response_only_simple(query, previous_context=''):
    # Construct the prompt
    prompt = f"{previous_context}<Human>: {query}\n<Assistant>: "
    encodings = trained_model_tokenizer(prompt, return_tensors='pt').to(device)

    with torch.inference_mode():
        outputs = trained_model.generate(
            input_ids=encodings.input_ids,
            attention_mask=encodings.attention_mask,
            max_length=1024,
            num_return_sequences=1,
            do_sample=True,
            temperature=0.7,
            top_p=0.9
        )

    full_response = trained_model_tokenizer.decode(outputs[0], skip_special_tokens=True)

    # Extracting only the first assistant's response
    # Split by "<Assistant>:" and "<Human>:", and take the first chunk after "<Assistant>:"
    response_parts = full_response.split("<Assistant>:")
    if len(response_parts) > 1:
        first_response = response_parts[1].split("<Human>:")[0].strip()
    else:
        first_response = response_parts[0].strip()  # Fallback in case "<Assistant>:" is not found

    return first_response
```

Fig 22: Responses Generation Code



```
# Example usage
response = generate_first_response_only_simple("what is")
print("Assistant:", response)
```

```
The current implementation of Falcon calls `torch.scaled_dot_product_at
Assistant: ? Is there anything I can help you with?
```

```
# Example usage
response = generate_first_response_only_simple("what is 2 + 2")
print("Assistant:", response)
```

```
Assistant: 4
```

```
# Example usage
response = generate_first_response_only_simple("how is the weather today?")
print("Assistant:", response)
```

```
Assistant: it's a little cloudy, and the temperature is 10 degrees Celsius.
```

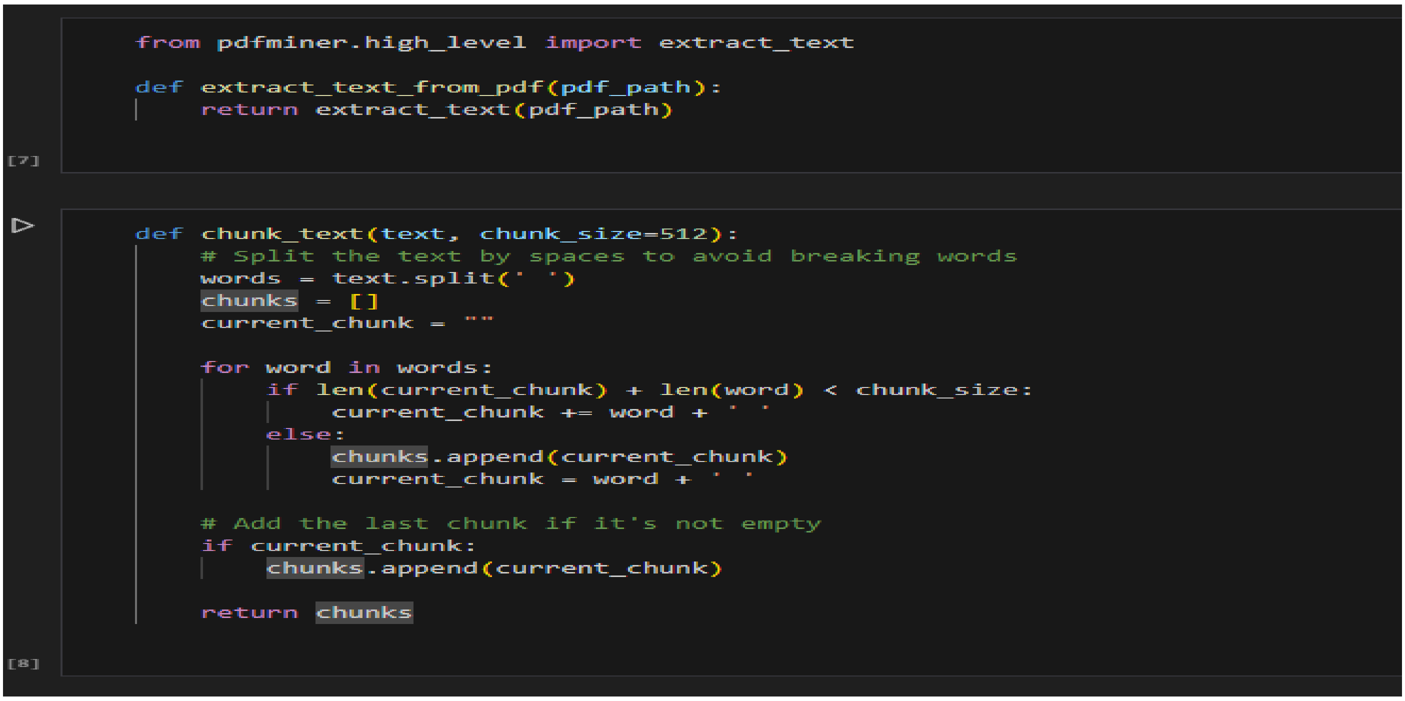
```
# Example usage
response = generate_first_response_only_simple("yellow or red")
print("Assistant:", response)
```

```
Assistant: @Human - you're the only one who can see it
```

Fig 23: Responses from Model

After loading the finetuned model, we then set up a simple code that allows us to perform inferences and get some responses from the model as shown in Fig.22 and Fig. 23.

Here we can observe that the model can provide normal responses when we ask it questions.

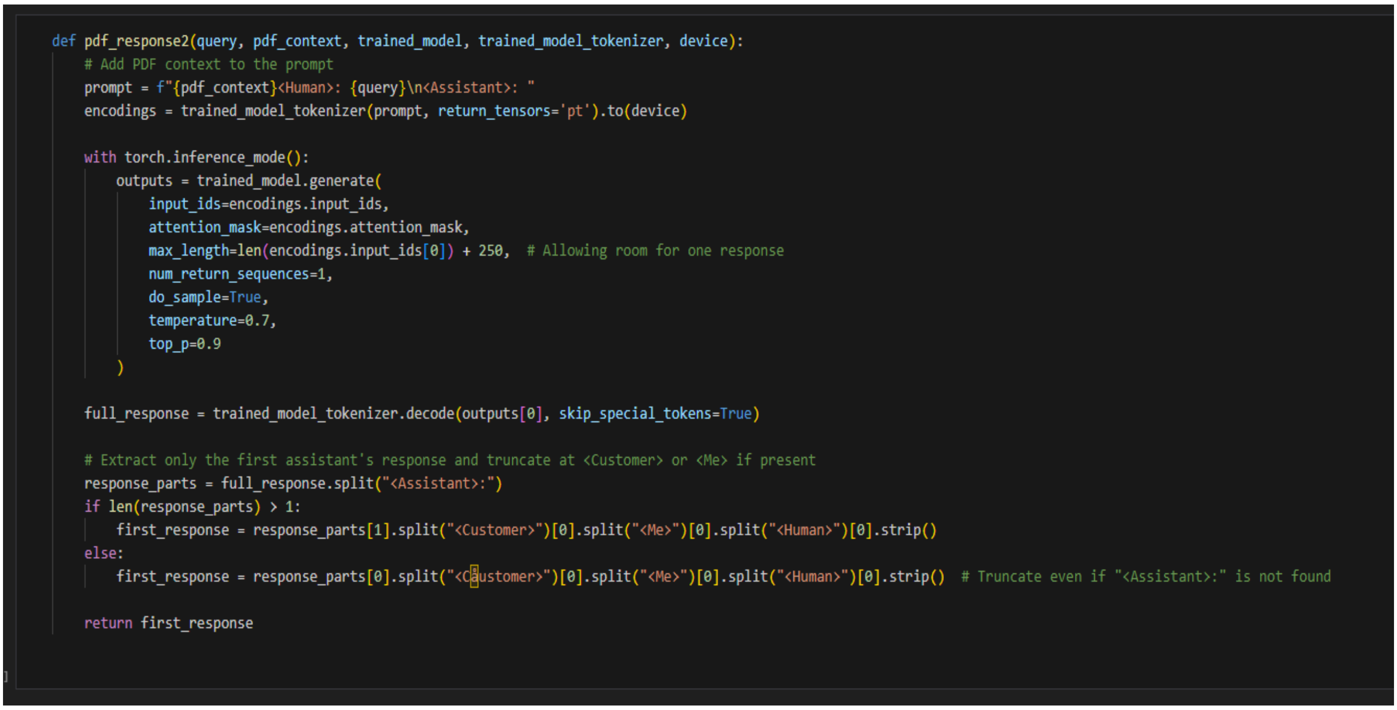


```
from pdfminer.high_level import extract_text

def extract_text_from_pdf(pdf_path):
    return extract_text(pdf_path)
```

```
def chunk_text(text, chunk_size=512):
    # Split the text by spaces to avoid breaking words
    words = text.split(' ')
    chunks = []
    current_chunk = ""

    for word in words:
        if len(current_chunk) + len(word) < chunk_size:
            current_chunk += word + ' '
        else:
            chunks.append(current_chunk)
            current_chunk = word + ' '

    # Add the last chunk if it's not empty
    if current_chunk:
        chunks.append(current_chunk)

    return chunks
```

Fig 24: PDF Processing

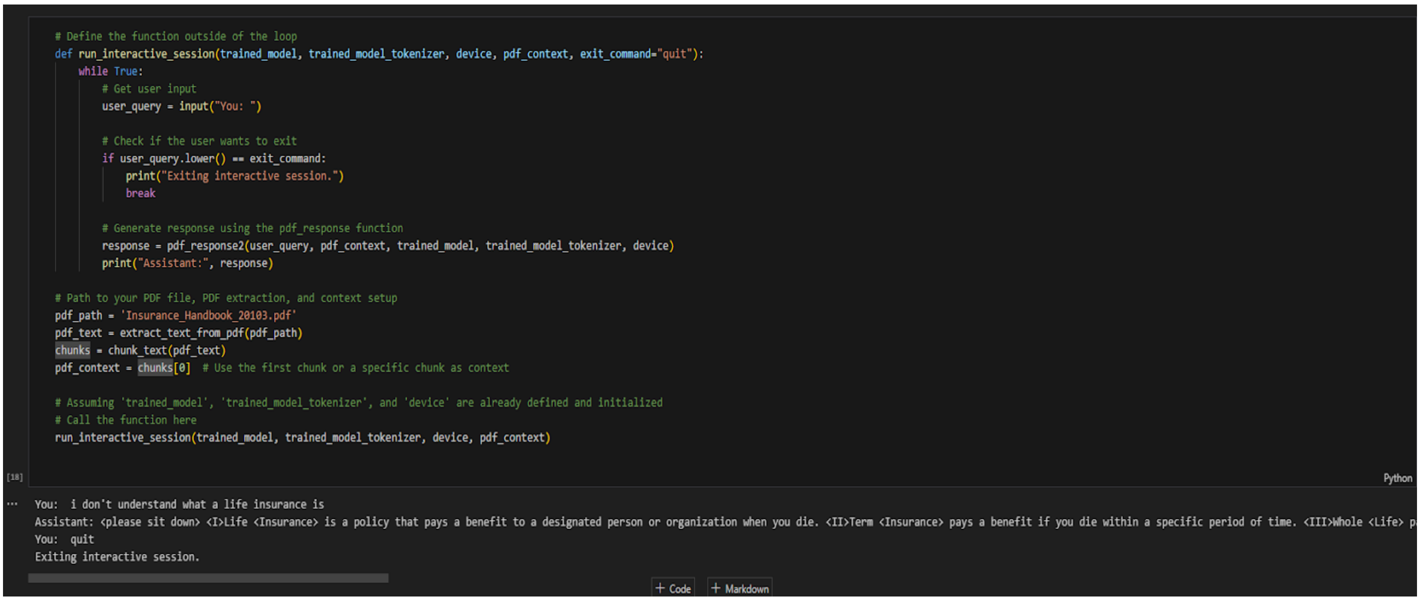
Lastly for this experiment, we tried to use Retrieval Augmented Generation so that the model can perform pdf querying. First up, we will use the chunk_text function to process the pdf in forms of chunks that is useful later for the model to provide relevant responses according to the information in the pdf as shown in Fig. 24 above.



```python
def pdf_response2(query, pdf_context, trained_model, trained_model_tokenizer, device):
    # Add PDF context to the prompt
    prompt = f"{pdf_context}<Human>: {query}\n<Assistant>: "
    encodings = trained_model_tokenizer(prompt, return_tensors='pt').to(device)

    with torch.inference_mode():
        outputs = trained_model.generate(
            input_ids=encodings.input_ids,
            attention_mask=encodings.attention_mask,
            max_length=len(encodings.input_ids[0]) + 250,  # Allowing room for one response
            num_return_sequences=1,
            do_sample=True,
            temperature=0.7,
            top_p=0.9
        )

    full_response = trained_model_tokenizer.decode(outputs[0], skip_special_tokens=True)

    # Extract only the first assistant's response and truncate at <Customer> or <Me> if present
    response_parts = full_response.split("<Assistant>:")
    if len(response_parts) > 1:
        first_response = response_parts[1].split("<Customer>")[0].split("<Me>")[0].split("<Human>")[0].strip()
    else:
        first_response = response_parts[0].split("<Customer>")[0].split("<Me>")[0].split("<Human>")[0].strip()  # Truncate even if "<Assistant>:" is not found

    return first_response
```

Fig 25: Code to Generate Responses from PDF

After the chunks have been processed, it is being used in this inference function that allows the model to reply with relevant information related to the pdf that is being uploaded as shown in Fig. 25.



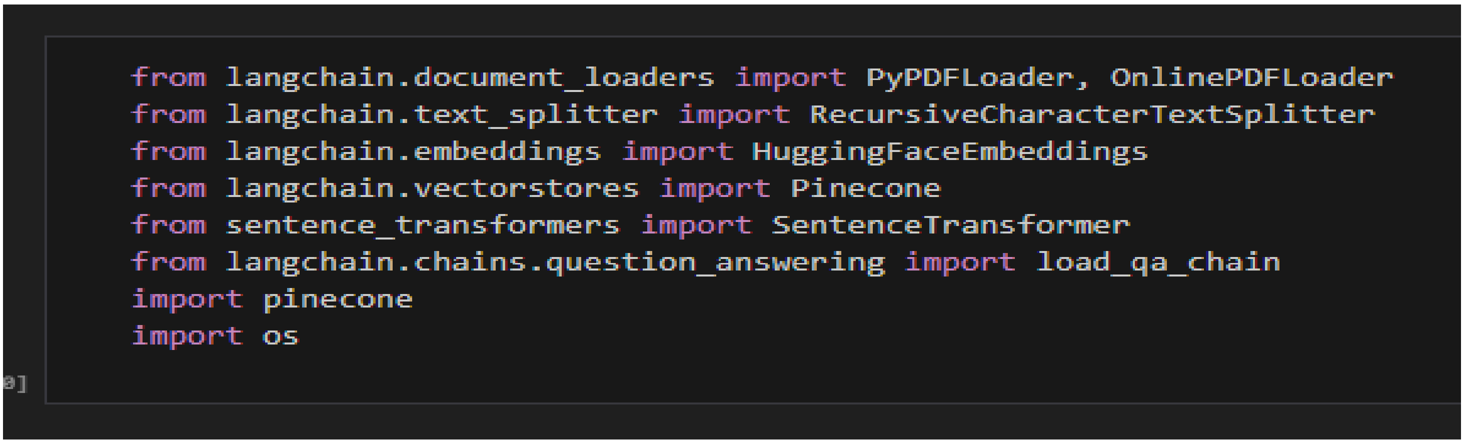
```python
# Define the function outside of the loop
def run_interactive_session(trained_model, trained_model_tokenizer, device, pdf_context, exit_command="quit"):
    while True:
        # Get user input
        user_query = input("You: ")

        # Check if the user wants to exit
        if user_query.lower() == exit_command:
            print("Exiting interactive session.")
            break

        # Generate response using the pdf_response function
        response = pdf_response2(user_query, pdf_context, trained_model, trained_model_tokenizer, device)
        print("Assistant:", response)

# Path to your PDF file, PDF extraction, and context setup
pdf_path = 'Insurance_Handbook_20103.pdf'
pdf_text = extract_text_from_pdf(pdf_path)
chunks = chunk_text(pdf_text)
pdf_context = chunks[0]  # Use the first chunk or a specific chunk as context

# Assuming 'trained_model', 'trained_model_tokenizer', and 'device' are already defined and initialized
# Call the function here
run_interactive_session(trained_model, trained_model_tokenizer, device, pdf_context)
```

```
You:  i don't understand what a life insurance is
Assistant: <please sit down> <I>Life <Insurance> is a policy that pays a benefit to a designated person or organization when you die. <II>Term <Insurance> pays a benefit if you die within a specific period of time. <III>Whole <Life> p
You:  quit
Exiting interactive session.
```

Fig 26: Simple Code to Test out Chatbot Feature

As we can observe from the response from the model above in Fig. 26, it is still acceptable but could be better. This occurrence is because the model is not properly fine-tuned with the relevant dataset and is not suitable for pdf querying, hence a future improvement would be to fine-tune it with the relevant dataset to get a more natural and accurate response without the need for pdf or any external resources.

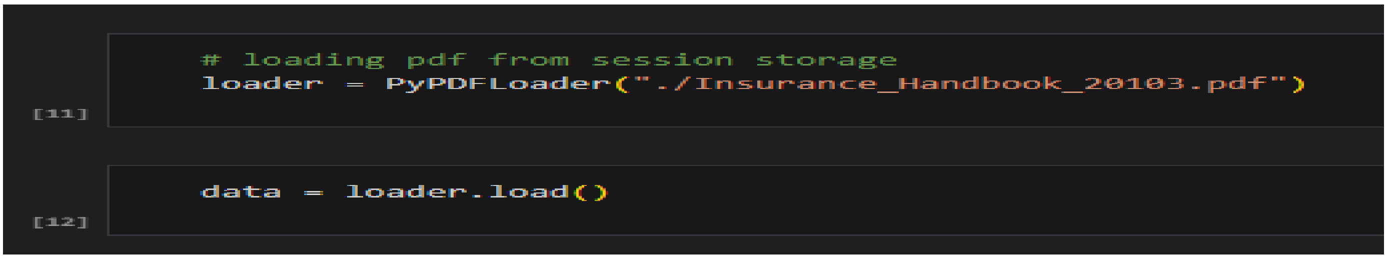
D. Data Preparation for Llama2B + Langchain



```
from langchain.document_loaders import PyPDFLoader, OnlinePDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Pinecone
from sentence_transformers import SentenceTransformer
from langchain.chains.question_answering import load_qa_chain
import pinecone
import os
```

Fig 27: Importing Required Libraries

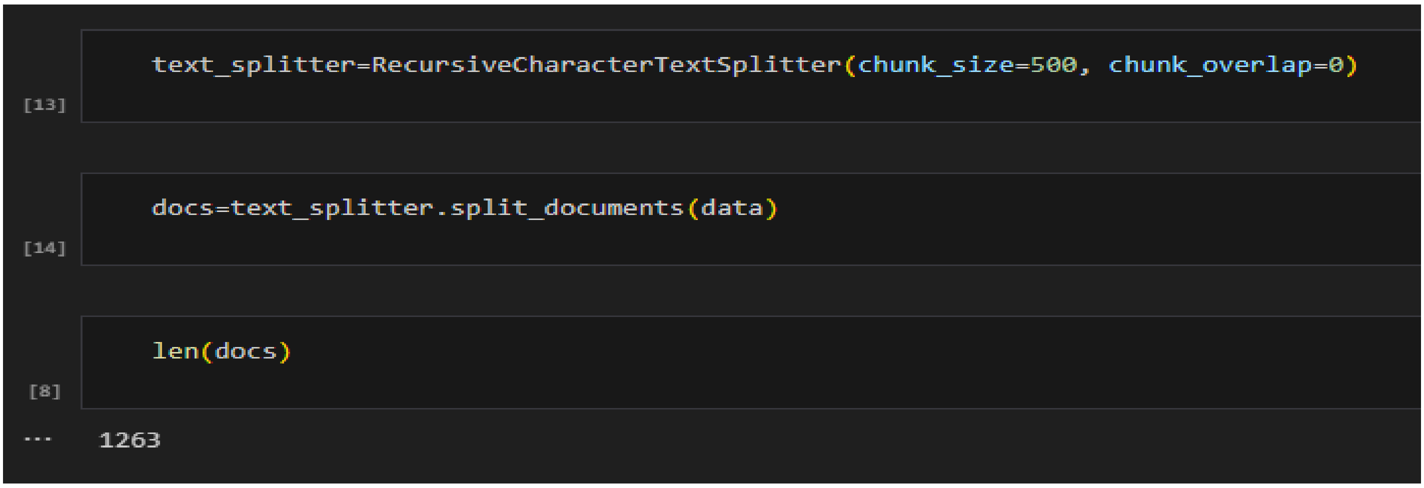
First, in Fig. 27, we will import all necessary libraries to get this model running together with the use of langchain.



```
[11]  # loading pdf from session storage
      loader = PyPDFLoader("./Insurance_Handbook_20103.pdf")

[12]  data = loader.load()
```

Fig 28: Loading PDF

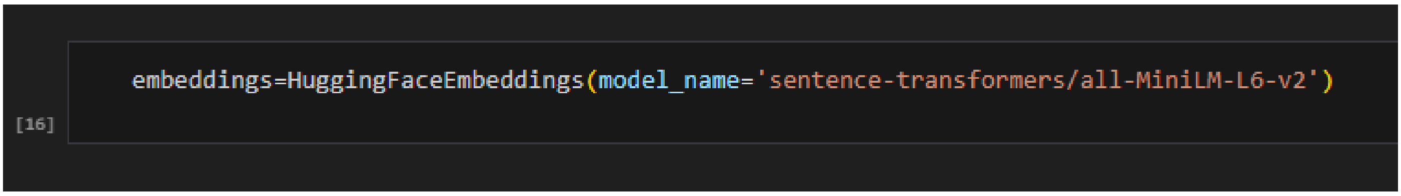
The pdf is then loaded from the local storage as shown in Fig. 28.



```
[13]  text_splitter=RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=0)

[14]  docs=text_splitter.split_documents(data)

[8]   len(docs)

...   1263
```

Fig 29: PDF Processing

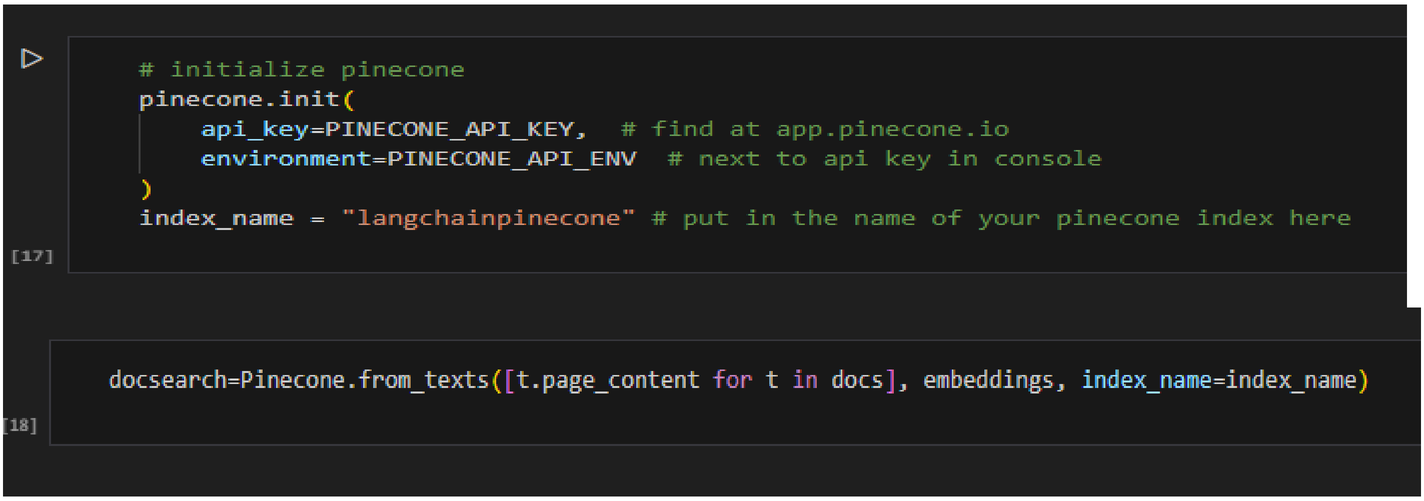
In Fig. 29, the code above will then process the PDF characters into chunks for embedding later.



```
[16]  embeddings=HuggingFaceEmbeddings(model_name='sentence-transformers/all-MiniLM-L6-v2')
```

Fig 30: Downloading Embeddings

The code in Fig. 30 above shows that we have to download the embeddings first which is “all-MiniLM-L6-v2”.



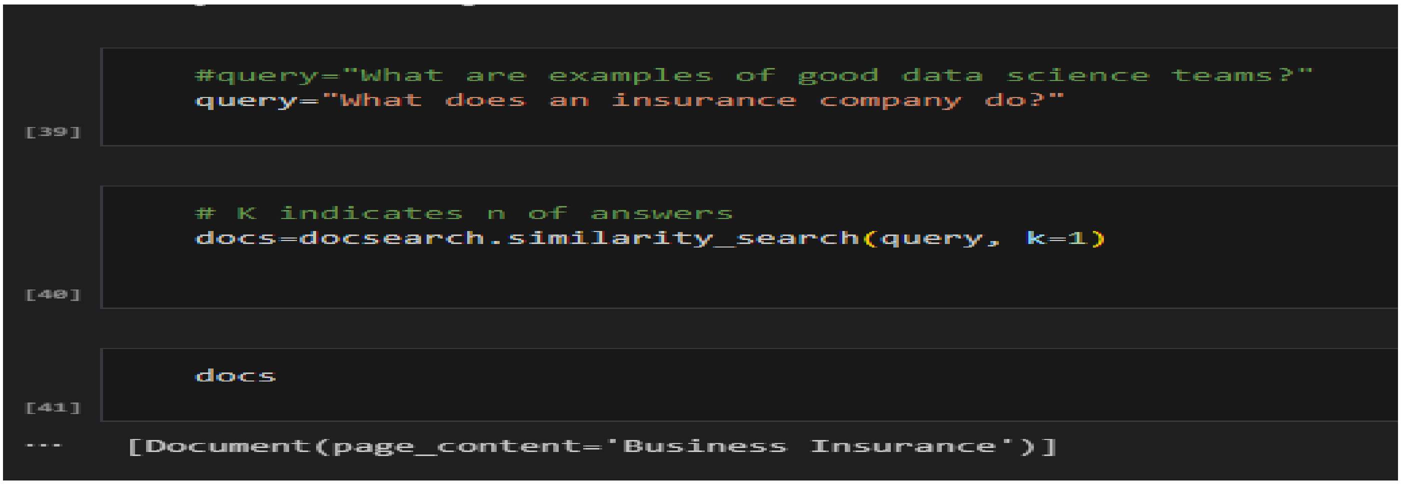
```
# initialize pinecone
pinecone.init(
    api_key=PINECONE_API_KEY,  # find at app.pinecone.io
    environment=PINECONE_API_ENV  # next to api key in console
)
index_name = "langchainpinecone" # put in the name of your pinecone index here
```
[17]

```
docsearch=Pinecone.from_texts([t.page_content for t in docs], embeddings, index_name=index_name)
```
[18]

Fig 31: Initialising Pinecone and setting up the environment

Then the pinecone is initialized so that we can create embeddings for each of the text chunks from the previous process as shown in Fig. 31.

E. Llama2 + Langchain Inferencing and RAG



```
#query="What are examples of good data science teams?"
query="What does an insurance company do?"
```
[39]

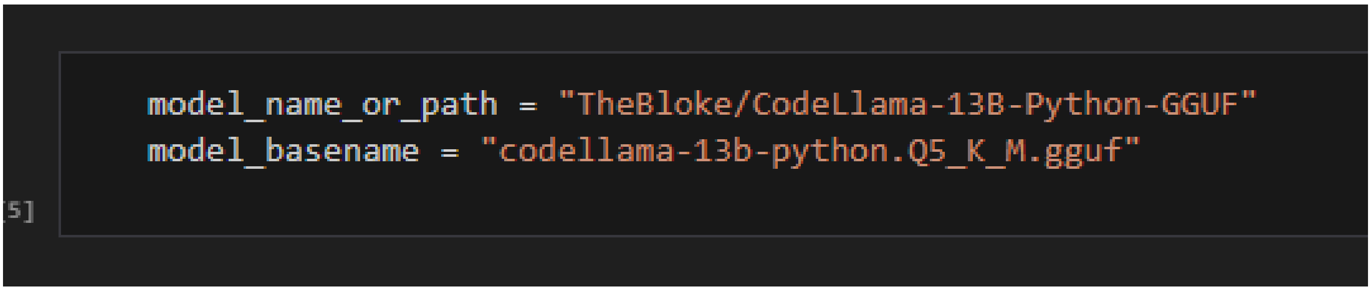
```
# K indicates n of answers
docs=docsearch.similarity_search(query, k=1)
```
[40]

```
docs
```
[41]

```
[Document(page_content='Business Insurance')]
```

Fig 32: Testing Embeddings Function

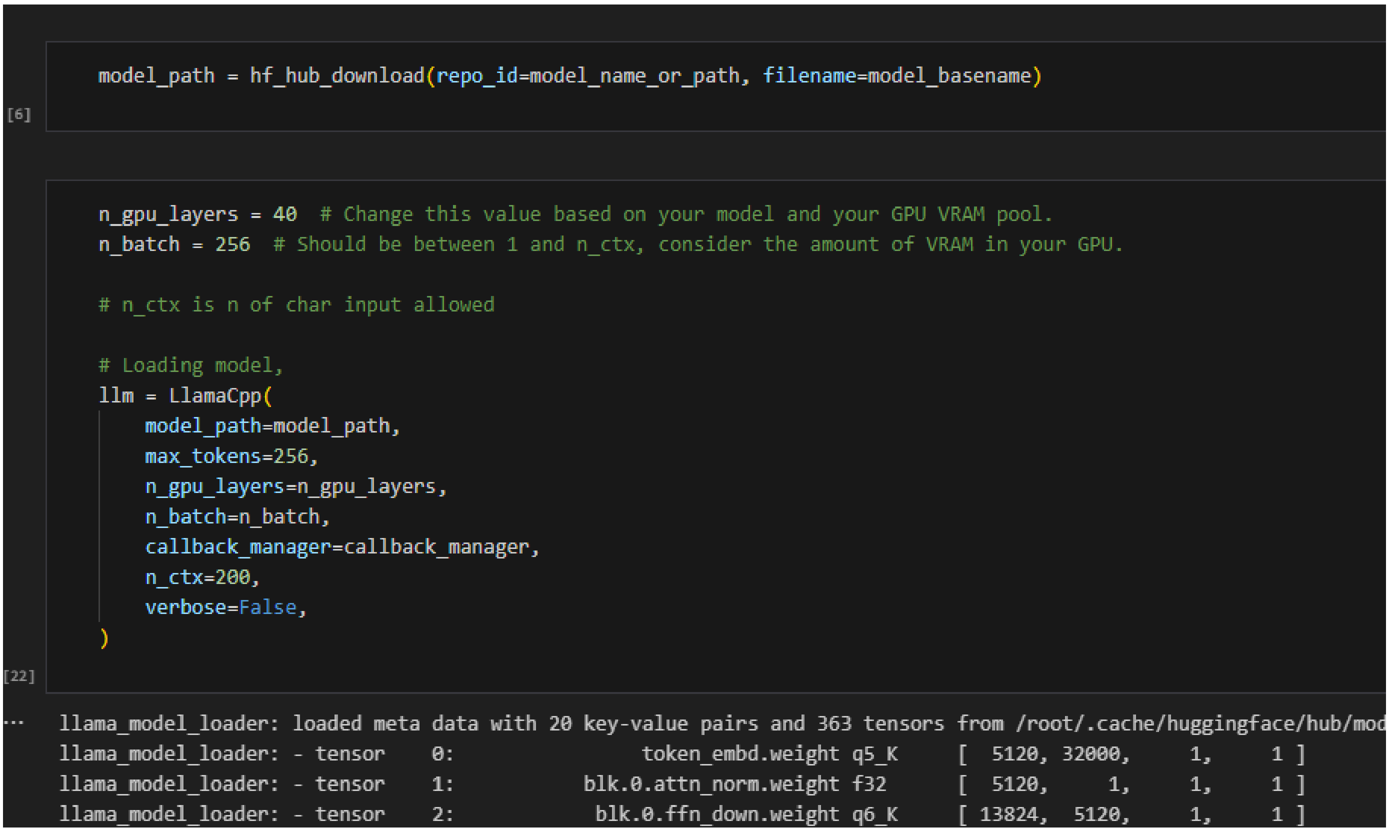
Referring to Fig. 32, after the embeddings have been created, we can now try to provide a query and observe its response. We will first test it by using the similarity search to validate if the docsearch is searching for relevant information from the document based on the query that we have provided. In this case, it has succeeded.



```
model_name_or_path = "TheBloke/CodeLlama-13B-Python-GGUF"
model_basename = "codellama-13b-python.Q5_K_M.gguf"
```
[5]

Fig 33: Importing Llama2 Model

Now we will load the quantized Llama model for inferencing and pdf querying shown clearly in Fig. 33.



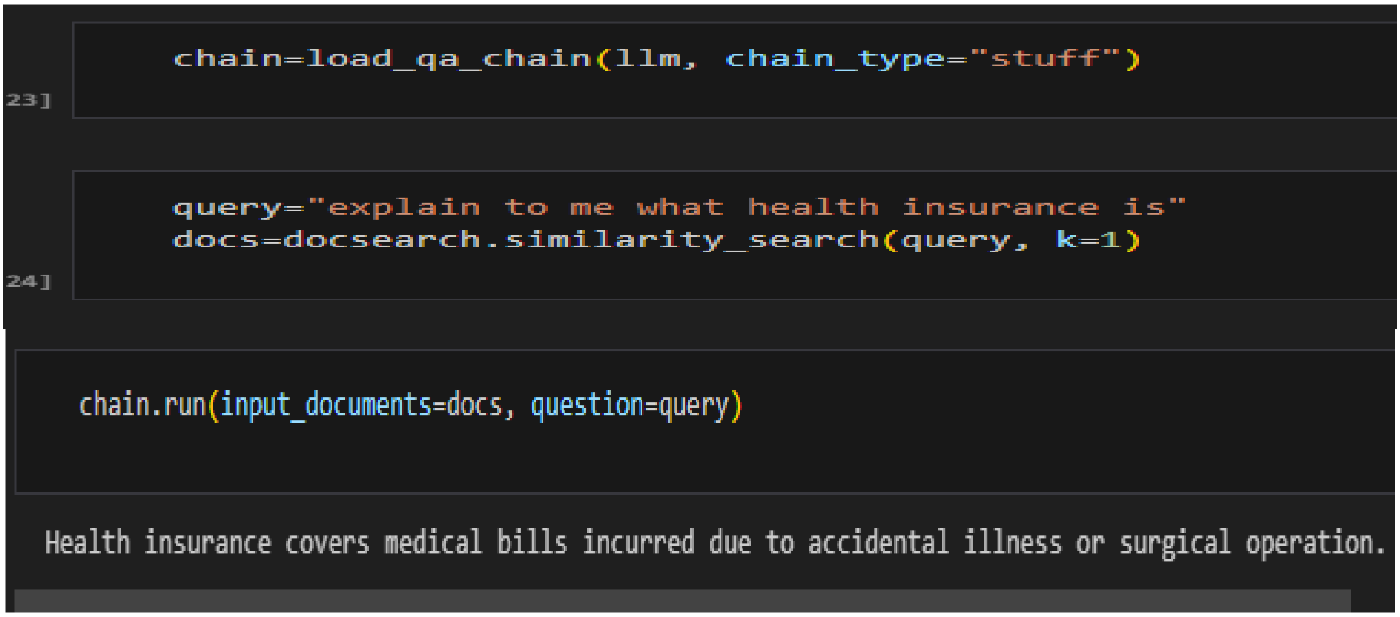
```
model_path = hf_hub_download(repo_id=model_name_or_path, filename=model_basename)
```
[6]

```
n_gpu_layers = 40  # Change this value based on your model and your GPU VRAM pool.
n_batch = 256  # Should be between 1 and n_ctx, consider the amount of VRAM in your GPU.

# n_ctx is n of char input allowed

# Loading model,
llm = LlamaCpp(
    model_path=model_path,
    max_tokens=256,
    n_gpu_layers=n_gpu_layers,
    n_batch=n_batch,
    callback_manager=callback_manager,
    n_ctx=200,
    verbose=False,
)
```
[22]

```
llama_model_loader: loaded meta data with 20 key-value pairs and 363 tensors from /root/.cache/huggingface/hub/mod
llama_model_loader: - tensor    0:                token_embd.weight q5_K     [  5120, 32000,     1,     1 ]
llama_model_loader: - tensor    1:           blk.0.attn_norm.weight f32      [  5120,     1,     1,     1 ]
llama_model_loader: - tensor    2:            blk.0.ffn_down.weight q6_K     [ 13824,  5120,     1,     1 ]
```

Fig 34: Setting Parameters for Llama2

Fig. 34 above shows that the parameters are being set up to control how the response will be given by the model where the n_ctx is the number of character inputs allowed into the model during inference. N_gpu_layers and n_batch determine how complex the model can be processed to generate a more accurate answer according to the questions asked, but it will then require more computational power.



```
chain=load_qa_chain(llm, chain_type="stuff")
```
23]

```
query="explain to me what health insurance is"
docs=docsearch.similarity_search(query, k=1)
```
24]

```
chain.run(input_documents=docs, question=query)
```

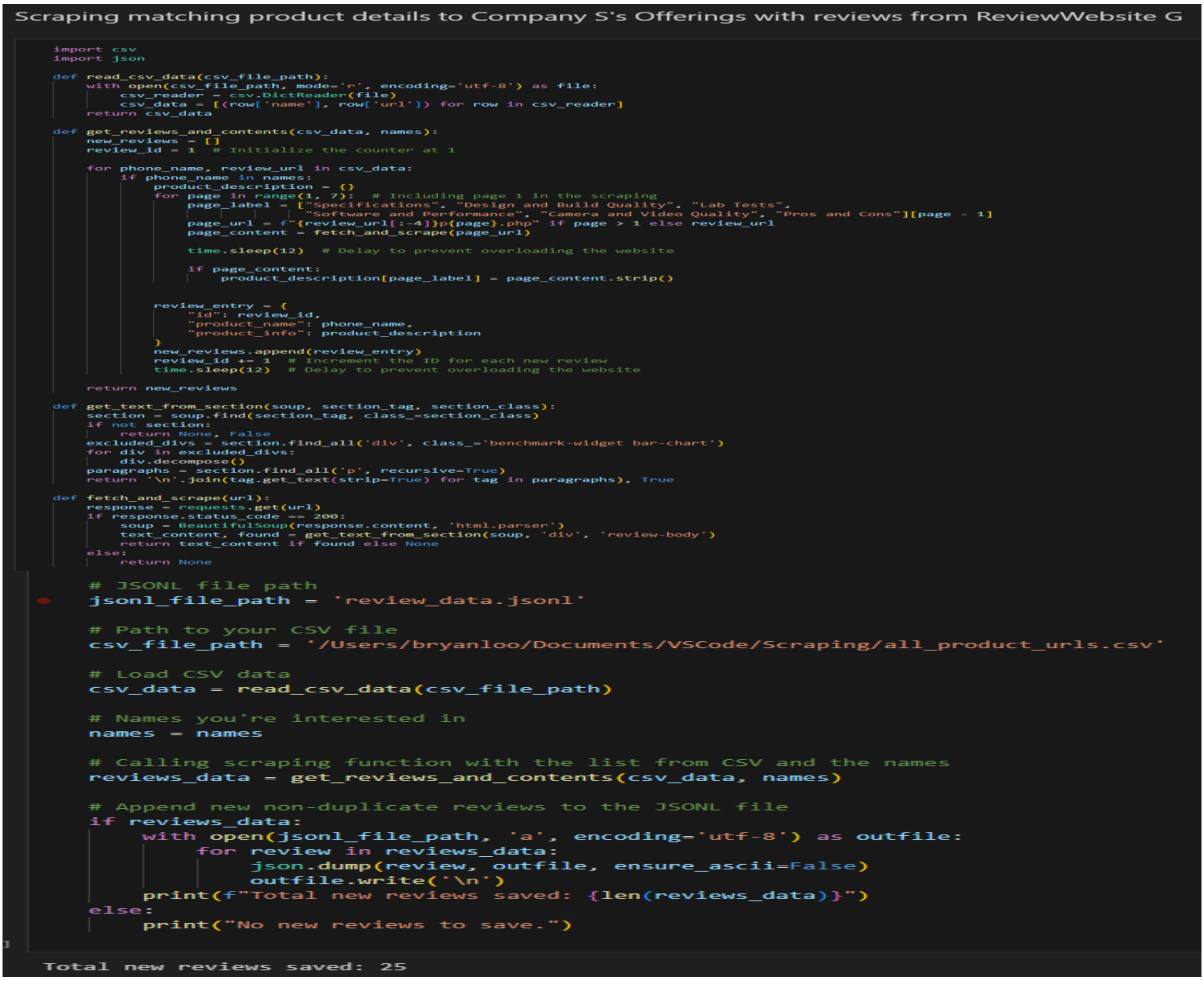
```
Health insurance covers medical bills incurred due to accidental illness or surgical operation.
```

Fig 35: Using Langchain for Inference

With the help of the langchain library, we can enter our input and observe that the model can provide a normal relevant response to us as demonstrated in Fig. 35.

F. *Sraping Script*

Fig. 36 below shows the code used for scrapping purposes.



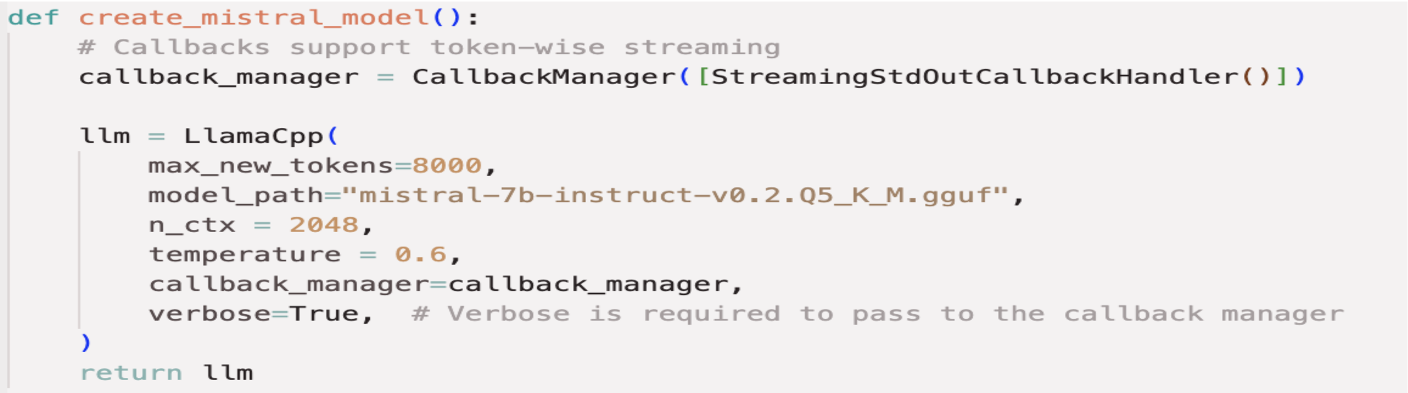
Scraping matching product details to Company S's Offerings with reviews from ReviewWebsite G

```
import csv
import json

def read_csv_data(csv_file_path):
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        csv_reader = csv.DictReader(file)
        csv_data = [(row['name'], row['url']) for row in csv_reader]
    return csv_data

def get_reviews_and_contents(csv_data, names):
    new_reviews = []
    review_id = 1  # Initialize the counter at 1

    for phone_name, review_url in csv_data:
        if phone_name in names:
            product_description = {}
            for page in range(1, 7):  # Including page 1 in the scraping
                page_label = ["Specifications", "Design and Build Quality", "Lab Tests",
                              "Software and Performance", "Camera and Video Quality", "Pros and Cons"][page - 1]
                page_url = f"{review_url[:-4]}p{page}.php" if page > 1 else review_url
                page_content = fetch_and_scrape(page_url)

                time.sleep(12)  # Delay to prevent overloading the website

                if page_content:
                    product_description[page_label] = page_content.strip()


            review_entry = {
                "id": review_id,
                "product_name": phone_name,
                "product_info": product_description
            }
            new_reviews.append(review_entry)
            review_id += 1  # Increment the ID for each new review
            time.sleep(12)  # Delay to prevent overloading the website

    return new_reviews

def get_text_from_section(soup, section_tag, section_class):
    section = soup.find(section_tag, class_=section_class)
    if not section:
        return None, False
    excluded_divs = section.find_all('div', class_='benchmark-widget bar-chart')
    for div in excluded_divs:
        div.decompose()
    paragraphs = section.find_all('p', recursive=True)
    return '\n'.join(tag.get_text(strip=True) for tag in paragraphs), True

def fetch_and_scrape(url):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        text_content, found = get_text_from_section(soup, 'div', 'review-body')
        return text_content if found else None
    else:
        return None
# JSONL file path
jsonl_file_path = 'review_data.jsonl'

# Path to your CSV file
csv_file_path = '/Users/bryanloo/Documents/VSCode/Scraping/all_product_urls.csv'

# Load CSV data
csv_data = read_csv_data(csv_file_path)

# Names you're interested in
names = names

# Calling scraping function with the list from CSV and the names
reviews_data = get_reviews_and_contents(csv_data, names)

# Append new non-duplicate reviews to the JSONL file
if reviews_data:
    with open(jsonl_file_path, 'a', encoding='utf-8') as outfile:
        for review in reviews_data:
            json.dump(review, outfile, ensure_ascii=False)
            outfile.write('\n')
    print(f"Total new reviews saved: {len(reviews_data)}")
else:
    print("No new reviews to save.")
```

```
Total new reviews saved: 25
```

Fig 36: Scrapping Script

G. *Mistral7B Setup*

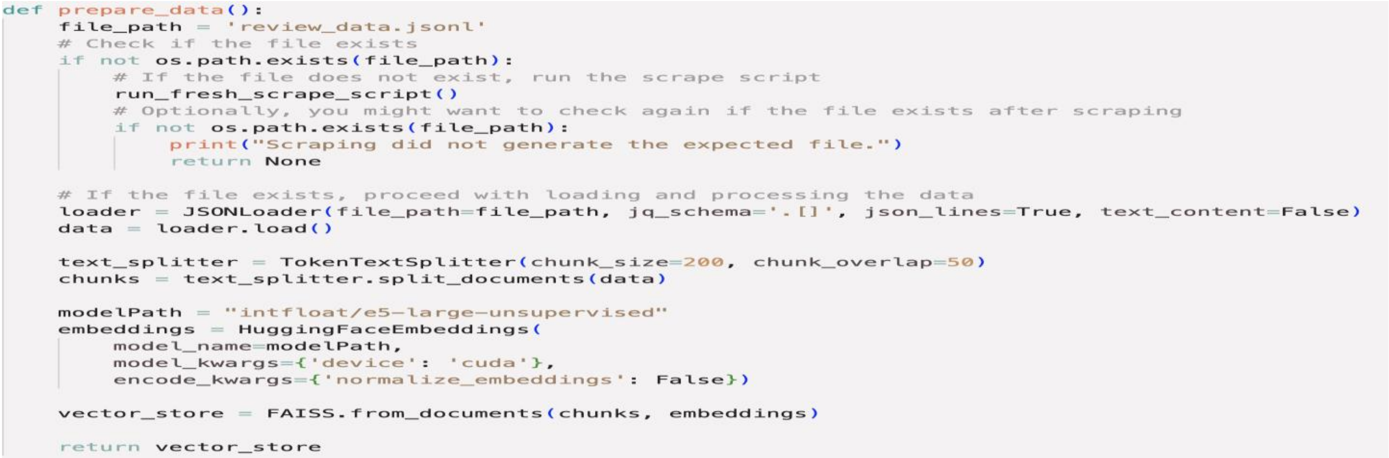


```
def create_mistral_model():
    # Callbacks support token-wise streaming
    callback_manager = CallbackManager([StreamingStdOutCallbackHandler()])

    llm = LlamaCpp(
        max_new_tokens=8000,
        model_path="mistral-7b-instruct-v0.2.Q5_K_M.gguf",
        n_ctx = 2048,
        temperature = 0.6,
        callback_manager=callback_manager,
        verbose=True,  # Verbose is required to pass to the callback manager
    )
    return llm
```

Fig 37: Setting Up LLM Model

By referring to Fig. 37, we configured the max_new_tokens and set it to 8000 to ensure the generation of comprehensive responses without prematurely cutting off, thereby minimizing the risk of incomplete answers. The context window (n_ctx) is extended to 2048 tokens, which allows the model to consider more extensive input for better-informed and contextually relevant outputs. A temperature setting of 0.6 strikes a balance between predictability and creativity, producing responses that are reliable yet nuanced. The inclusion of a callback_manager with a StreamingStdOutCallbackHandler facilitates real-time observation of responses for debugging purposes. Lastly, verbose=True is enabled to provide detailed logs during operation, essential for monitoring the model's internal processes and fine-tuning its performance for optimal interaction quality. These settings collectively aim to enhance the user experience by ensuring that the chatbot can handle intricate queries with detailed and relevant replies.

H. Data Preprocess for RAG



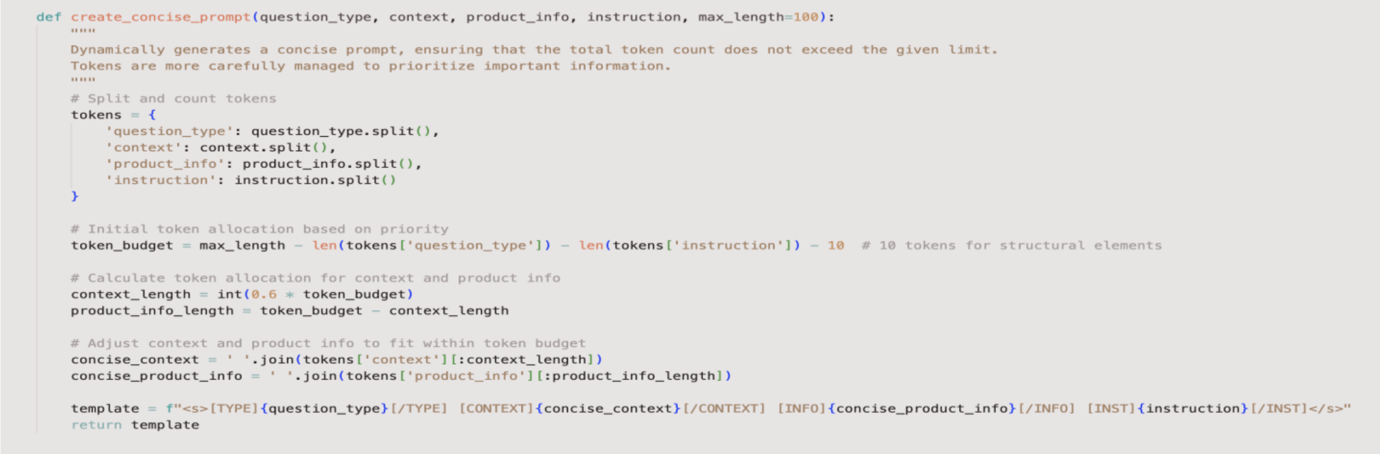
```
def prepare_data():
    file_path = 'review_data.jsonl'
    # Check if the file exists
    if not os.path.exists(file_path):
        # If the file does not exist, run the scrape script
        run_fresh_scrape_script()
        # Optionally, you might want to check again if the file exists after scraping
        if not os.path.exists(file_path):
            print("Scraping did not generate the expected file.")
            return None

    # If the file exists, proceed with loading and processing the data
    loader = JSONLoader(file_path=file_path, jq_schema='.[]', json_lines=True, text_content=False)
    data = loader.load()

    text_splitter = TokenTextSplitter(chunk_size=200, chunk_overlap=50)
    chunks = text_splitter.split_documents(data)

    modelPath = "intfloat/e5-large-unsupervised"
    embeddings = HuggingFaceEmbeddings(
        model_name=modelPath,
        model_kwargs={'device': 'cuda'},
        encode_kwargs={'normalize_embeddings': False})

    vector_store = FAISS.from_documents(chunks, embeddings)

    return vector_store
```

Fig 38: Data Preparation Code

Fig. 38 above shows the code used to prepare our data for RAG. This process begins with text splitting, where the data is divided into manageable pieces or 'chunks' that the model can easily process. In the context of RAG, which combines retrieval from a database with a generative language model, the ability to pinpoint relevant information rapidly is crucial. By breaking down the text into smaller segments, we facilitate a more precise and efficient retrieval process. This granularity ensures that the model can focus on the most relevant sections of text when generating a response to a query.

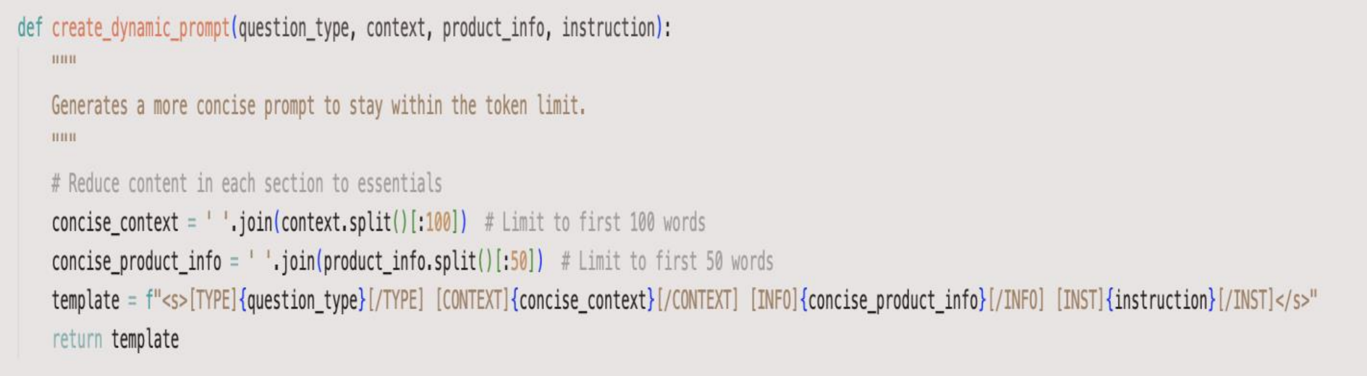
Once the text is split, chunking comes into play. Each chunk, often a paragraph or a set of sentences containing a complete thought, is then encoded into vectors using embeddings. These embeddings capture the semantic meaning of the text and allow the model to perform mathematical operations on the textual data. This step is vital because it transforms the text into a format that the machine learning model can understand and manipulate. Embedding the chunks into a vector space creates a searchable index that the RAG model can query to find the most relevant information for any given input. This process is critical for the model's ability to generate accurate, informative, and contextually relevant responses based on the input data it retrieves.

I. LLM Chaining and Output Translation Code



```
def create_concise_prompt(question_type, context, product_info, instruction, max_length=100):
    """
    Dynamically generates a concise prompt, ensuring that the total token count does not exceed the given limit.
    Tokens are more carefully managed to prioritize important information.
    """
    # Split and count tokens
    tokens = {
        'question_type': question_type.split(),
        'context': context.split(),
        'product_info': product_info.split(),
        'instruction': instruction.split()
    }

    # Initial token allocation based on priority
    token_budget = max_length - len(tokens['question_type']) - len(tokens['instruction']) - 10  # 10 tokens for structural elements

    # Calculate token allocation for context and product info
    context_length = int(0.6 * token_budget)
    product_info_length = token_budget - context_length

    # Adjust context and product info to fit within token budget
    concise_context = ' '.join(tokens['context'][:context_length])
    concise_product_info = ' '.join(tokens['product_info'][:product_info_length])

    template = f"<s>[TYPE]{question_type}[/TYPE] [CONTEXT]{concise_context}[/CONTEXT] [INFO]{concise_product_info}[/INFO] [INST]{instruction}[/INST]</s>"
    return template
```

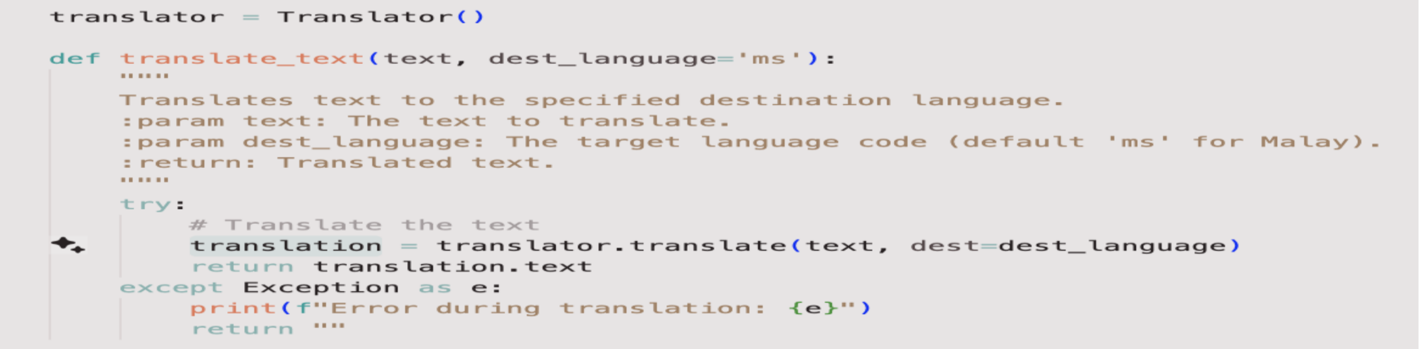
Fig 39: Function to Create Concise Prompt for LLM Chaining



```
def create_dynamic_prompt(question_type, context, product_info, instruction):
    """

    Generates a more concise prompt to stay within the token limit.
    """

    # Reduce content in each section to essentials
    concise_context = ' '.join(context.split()[:100])  # Limit to first 100 words
    concise_product_info = ' '.join(product_info.split()[:50])  # Limit to first 50 words
    template = f"<s>[TYPE]{question_type}[/TYPE] [CONTEXT]{concise_context}[/CONTEXT] [INFO]{concise_product_info}[/INFO] [INST]{instruction}[/INST]</s>"
    return template
```

Fig 40: Function to Create Dynamic Prompt for Final Output

After the user provides an input, the system will first classify the query before creating a custom prompt for generating the first response. A token limit is imposed to prevent the response from exceeding the context window for LLM Chaining later. From Fig. 39, after creating the first response using the template created by the create_concise_prompt() function, we then pass the output as an input for the next response generation to further refine the final output. This is done using another prompt template with the create_dynamic_prompt() function with minimal token limitation to ensure that it generates what is needed, shown in Fig. 40.



```
translator = Translator()

def translate_text(text, dest_language='ms'):
    """
    Translates text to the specified destination language.
    :param text: The text to translate.
    :param dest_language: The target language code (default 'ms' for Malay).
    :return: Translated text.
    """
    try:
        # Translate the text
        translation = translator.translate(text, dest=dest_language)
        return translation.text
    except Exception as e:
        print(f"Error during translation: {e}")
        return ""
```

Fig 41: English to Bahasa Malayu Translation Code

Lastly, the final output is then displayed together with its translated version with the code snippet in Fig. 41.