

**Enhancing Machine Learning Comprehension: A System for Reading and
Understanding Machine Learning Articles**

Nikhil Mylarusetty
Department of Applied Data Science, San Jose State University

DATA 298A: MSDA Project 1

Dr. Ming Hwa Wang

April 30, 2024

Abstract

Abstract

In today's rapidly evolving landscape of Machine Learning (ML), the volume of academic papers being published daily is staggering, averaging around 100 papers globally. This translates to approximately 36,500 papers per year, each containing valuable insights, innovations, and advancements in ML research. The goal is to establish a user-friendly interface having a focus on improving the efficiency and effectiveness of analyzing the machine learning (ML) research papers. The dataset consists of the papers from the arXiv database repository, which will serve as the foundation for subsequent analysis and model training. Moreover, another form of data is used which is a QASPER (questionnaire) dataset. To streamline data storage and retrieval for quicker and more effective searches and analyses, the collected research papers are seamlessly integrated into the Pinecone vector database. To enhance natural language processing capabilities, a Large Language Model (LLM) is created, Utilizing llama 2, Mistral 7B and retrieval augmented generation which will be trained and fine-tune thoroughly to facilitate precise comprehension and generation, improving user assistance. The efficacy of the Large Language Model is rigorously evaluated through EVALS metrics, including context relevancy, faithfulness, answer relevancy, context recall, and harmfulness, ensuring coherent and contextually relevant responses. Users can engage with the material through a questionnaire, seeking clarification and gaining deeper insights into technical content, especially in navigating the complexities of ML papers. Notably, the RAG with Custom Finetuned LLaMa model demonstrated the highest faithfulness (0.9074), while the RAG with Base Mistral model achieved the highest context relevancy score (0.5238). Conversely, the RAG with Custom FT Mistral model exhibited the lowest faithfulness (0.2441) among the evaluated models.

1. Introduction

1.1 Project Background and Summary

The “Reading Understanding System for Machine Learning Research Papers” project aims to enhance the accessibility and comprehension of machine learning research papers, which are often abundant and complex. The motivation behind this initiative is the growing need for researchers, students, and professionals to quickly access relevant information from a vast corpus of research papers. Large Language learning models (LLMs) have the potential to significantly streamline this process.

In this project, a comprehensive dataset is collected from the arXiv database, encompassing a wide array of machine learning research papers. These papers are then processed into smaller text chunks, enabling efficient handling and analysis for Retrieval Augmented Generation technique for LLM Model for machine learning Domain research papers. However, the heart of the system lies in the process of fine-tuning LLM where the dataset for this is collected from QASPER Dataset.

The key innovation lies in the conversion of these text chunks into embeddings, effectively creating a numerical representation of the textual content. This step ensures that the information extracted is not only manageable but also carries semantic meaning.

The vector database that stores these embeddings, making it feasible to search for semantically related content based on user queries. When a user submits a question or prompt, it is transformed into a corresponding vector. A semantic search is executed within the vector database, retrieving text chunks and research papers that closely match the user's query. Both the query embedding and the semantic embedding is fed to the LLM Model and the output is generated.

However, the core of the system is to further enhance the quality of the responses, a fine-tuned LLama model and Mistral 7B which are employed to be trained on QASPER

Dataset. This model has been meticulously optimized through the utilization of PEFT (Parameter Efficient Fine Tuning) which involves LoRA and QLoRA, a question-answering system that leverages large language models. Fine-tuning is the linchpin of this system, as it adapts the model to the specific task of comprehending machine learning research papers.

By integrating semantic search with the fine-tuned LLama model and Mistral 7B, the system can deliver accurate and comprehensive results for understanding machine learning papers. This synergy ensures that the system not only provides relevant content but also does so in a manner that is coherent and understandable.

The intended contributions of this project are manifold. Firstly, it aims to provide a practical solution for researchers and practitioners to navigate the vast landscape of machine learning literature efficiently. Secondly, it facilitates better comprehension of research papers, aiding in the dissemination of knowledge within the machine learning community. Lastly, the combination of semantic search and a fine-tuned LLama model and also Mistral 7B models, achieved through rigorous fine-tuning, promises to significantly improve the accuracy and relevance of the results obtained from the vast collection of research papers and QASPER Dataset.

In the end, our project fills a crucial need in the machine learning research area by providing a potent tool to academics and enthusiasts who want to keep up with the most recent advancements and insights in the industry. It has the ability to fundamentally alter how we access, comprehend, and make use of research articles in the machine learning environment by utilizing the capabilities of machine learning and the crucial process of fine-tuning.

1.2 Project requirements

The project aims to create an ML research paper comprehension interface with a focus on fine-tuning AI capabilities. Technical requirements include high-performance

GPUs, computing resources with adequate RAM and CPU cores, frameworks like TensorFlow or PyTorch, access to the Arxiv database, integration with the Pinecone vector database, a pretrained AI models such as Llama 2 7B and Mistral 7B, Langchain or similar blockchain integration for security, a token-based reward system, and a comprehension questionnaire feature. These technical elements are essential for developing an efficient system that enhances the understanding of ML research papers through advanced AI-driven functionalities

Data Requirements

The success of our project relies on the availability and management of key datasets. To facilitate efficient paper retrieval and analysis, it will require access to a substantial dataset of machine learning (ML) research papers sourced from the Arxiv database. This dataset will serve as the foundation for subsequent analysis and model training. In addition to this, the QASPER Dataset is also used for performing tuning the parameters. Additionally, seamless integration with the Pinecone vector database is essential for quick and effective data storage and retrieval. This project will also need a robust and reliable source of pre-trained models, such as Llama 2 and Mistral 7B, for the development and fine-tuning of our Large Language Model (LLM). Furthermore, the system must collect and store user interaction data, including queries and feedback, to continually improve its performance. Lastly, the implementation of Langchain technology is necessary to ensure data security, decentralization, and integrity. This combination of data sources and technologies will underpin the functionality and intelligence of our system.

Functional Requirements

The project's primary aim is to craft a user-friendly interface tailored for the enhancement of efficiency and effectiveness in the comprehension of machine learning (ML) research papers. The central emphasis lies in the fine-tuning of AI capabilities to achieve this

goal. To make this vision a reality, certain functional requirements must be met. Firstly, it should efficiently retrieve relevant ML research papers, and this efficiency will be measured by the average time taken to retrieve a paper based on user queries. User interface (UI) usability is another critical aspect, and its success will be measured through user satisfaction scores obtained via surveys and usability testing.

The system's natural language processing (NLP) performance, fine-tuned through continuous learning, will be evaluated using metrics such as perplexity, BLEU score, and ROUGE score, in addition to user feedback on response quality. To ensure data integration and storage efficiency, the integration with the Pinecone vector database will be assessed based on database query response time and storage utilization. Security, decentralization, and transparency are paramount, necessitating the implementation of Langchain and a token-based reward system to incentivize user participation, both of which will be monitored and evaluated.

Lastly, the system's efficacy in assisting users in comprehending ML papers will be gauged through user participation in comprehension questionnaires and feedback on its effectiveness in enhancing comprehension. The fine-tuned AI will play a crucial role in providing coherent and contextually relevant responses, moving beyond mere summarization to foster a profound understanding of the technical content. This AI-driven approach ensures that users can navigate the complexities of ML papers with the highest degree of precision and accuracy.

AI Requirements

The utilization of artificial intelligence (AI) will play an important role in improving the system's capabilities and intelligence to accomplish this, a specific set of AI requirements has been defined. Semantic search functionality will be developed to enable the retrieval of papers based on context and meaning, as opposed to relying solely on keywords. The

effectiveness of semantic search will be measured through the system's ability to find relevant papers using semantic similarity metrics and user feedback. Furthermore, automatic paper summarization will be implemented to provide concise overviews of research papers, with the quality of summaries assessed using metrics like ROUGE and user satisfaction ratings. The AI will also be trained to answer user questions about paper content, and the accuracy of AI-generated answers will be evaluated through user feedback and performance metrics. The system will incorporate topic modeling and categorization to aid organization and search, with the accuracy of categorization and relevance of suggested categories measured. Personalized recommendations will be generated based on user interests, and the system's ability to adapt recommendations over time will be monitored. Additionally, natural language generation (NLG) capabilities will be utilized to generate explanations and contextual information, with clarity and usefulness of NLG-generated content evaluated. Performance monitoring and continuous learning mechanisms will be in place to track system performance and improvements over time. Finally, user profiling and multilingual support will enhance user experience and comprehension, with their effectiveness gauged through improved recommendations and user feedback.

1.3 Project Deliverables

The project encompasses multiple critical phases and deliverables. It begins with the Abstract, providing an overview of project objectives, scope, stakeholders, and a preliminary plan for achieving goals. This sets the high-level direction and strategy. The Work Breakdown Structure (WBS) then delves into the detailed project tasks, sub-tasks, dependencies, and timelines, acting as a roadmap for efficient resource allocation and progress tracking. The Introduction phase dives into project background, requirements, deliverables, and technology survey, along with a literature review of existing research. The

Data and Project Management Plan focus on data collection, storage, usage strategies, privacy, access, security considerations, and project management workflows.

Table 1
Project Deliverables

Chapters	Phase	Concept	Description	Due Date
Abstract	Abstract	Abstract Document	An overview of the project objectives, scope, stakeholders, and a preliminary plan for achieving project goals. This document outlines the high-level project direction and strategy.	09-13-2023
Chapter 1	Introduction	Project Background	A report that includes the project history, needs, deliverables, technology and solution survey, and a study of available research literature.	09-26-2023
Chapter 2	Work Breakdown Structure (WBS)	WBS Document	A detailed breakdown of project tasks, sub-tasks, dependencies, and timelines. The WBS serves as a roadmap for project management, enabling efficient allocation of resources and tracking progress.	09-21-2023
	Data and Project Management Plan	Data Management Plan	A document outlining data collection, storage, and usage strategies. It addresses data privacy, access, and security considerations, along with a project management plan defining roles and workflows.	10-05-2023
Chapter 3	Data Engineering	ML Research Papers Dataset and QASPER Dataset	The collection and preprocessing of a dataset from the Arxiv and QASPER is ready for model development. These datasets are crucial for AI training and validation.	10-10-2023
		Pinecone Integration Setup	Configuration and integration of the Pinecone vector database for efficient data storage and retrieval of research papers.	10-17-2023
		Langchain Blockchain Integration	Implementation of Langchain or similar blockchain technology for ensuring data security, integrity, and decentralization.	10-24-2023
Chapter 4	Model Development	Pretrained AI Model Integration	Integration of a pretrained AI model (e.g., GPT-3.5 Turbo) for fine-tuning and use within the system to enhance natural language processing capabilities.	11-7-2023
		Fine-tuned AI Model	The development and fine-tuning of the Large Language Model (LLM) for generating coherent and contextually relevant responses.	11-14-2023
		Evaluation	Evaluating the Models Based on the proposed evaluation metrics	11-21-2023
298A Research Presentation	Research Presentation		A comprehensive presentation summarizing the project's goals, methodologies, results, and conclusions. This presentation is intended for stakeholders and may include a live system demonstration.	12-06-2023
298A Project Report	Project Report		Generate a comprehensive and informative final report that offers a thorough summary of the study, encompassing all essential and pertinent details. This report should be both detailed and insightful, ensuring that readers gain a well-defined grasp of the research approach and outcomes.	12-05-2023

Data Engineering involves critical tasks like creating an ML research papers dataset for model development and integrating technologies like Pinecone for efficient data storage and Langchain blockchain for data security. The Model Development stage encompasses integrating and fine-tuning AI models like Llama 2 and Mistral 7B which are essential for enhancing natural language processing capabilities. Additionally, a token-based reward system is planned to incentivize user engagement. The Data Analytics and Intelligent System phase involve the development of an analytics dashboard for insights and integration of advanced AI capabilities like semantic search, summarization, question answering, and topic modeling. Finally, the System Evaluation and Visualization phase generates a comprehensive

report detailing evaluation metrics and visualizing results, essential for refining and improving the system based on user feedback and performance metrics.

1.4 Technical Survey

The paper by Cheng et al. (2023) describes a study that focuses on training large language models (LLMs) using domain-specific corpora to adapt them. The authors suggest a method for converting domain-specific raw corpora into reading comprehension texts, which will allow the model to learn domain knowledge and improve prompting ability through comprehension tests. This method has been found to consistently boost performance in biomedical, finance, and law sectors. The results show that the suggested strategy outperforms domain-specific models at significantly larger scales, such as the BloombergGPT-50B, and even outperforms universal benchmarks. The paper compares the proposed method, AdaptLLM, with other publicly available models in the mentioned domains, highlighting its effectiveness and efficiency in comparison to training from scratch. The conclusion emphasizes the potential of the proposed method to enhance general language models across multiple domains and encourages further exploration in adapting large language models using large-scale unsupervised corpora efficiently for specialized areas (Cheng, Huang, & Wei, 2023).

Another paper by De Curtò et al. (2023) introduces an innovative approach utilizing Large Language Models (LLMs) and Visual Language Models (VLMs) in combination with a detection pipeline to generate literary text descriptions of scenes captured by Unmanned Aerial Vehicles (UAVs). The aim is to achieve zero-shot semantic scene understanding, providing applications in areas such as the filming industry, theme parks, or advertisement sectors. The proposed methodology involves real-time narrative generation based on data collected by UAVs, combining image data and other environmental information, processed using Machine Learning (ML) algorithms. The process encompasses CLIP for image

captioning, YOLOv7 for object detection, and GPT-3 for enhancing textual descriptions. The system demonstrates potential for diverse applications, including education, journalism, entertainment, enabling immersive experiences and enhanced understanding of events and situations. The results showcase the effectiveness and efficiency of the proposed pipeline, presenting promising avenues for further integration, optimization, and domain-specific customization. Future enhancements could involve trajectory optimization, additional sensor integration, improved autonomy, real-time analysis, and domain-specific tailoring to extend the literary capabilities of autonomous systems like UAVs and self-driving cars.

In this paper, the authors Siriwardhana et al. (2023) introduce RAG-end2end, a novel approach for domain adaptation in Open-Domain Question Answering (ODQA) using the Retrieval Augmented Generation (RAG) model. They address the challenge of adapting RAG to specialized domains like healthcare and news, which are not well-suited for the standard Wikipedia-based external knowledge base used by RAG. RAG-end2end is designed to adapt RAG to domain-specific knowledge bases by jointly training the retriever and generator components and introducing an auxiliary training signal for statement reconstruction. This approach is evaluated on three domains (COVID-19, News, and Conversations) and outperforms various RAG model variants, showing significant improvements in Exact Match (EM), F1 score, and retrieval accuracy across all domains. Even when added to the original RAG model, the auxiliary signal for statement reconstruction enhances performance, particularly in the Conversation domain, where the EM score improves by 13 points. In summary, RAG-end2end demonstrates promising results, leveraging joint training, domain-specific data, and statement reconstruction to enhance adaptability in ODQA for specialized domains

The paper by Wang et al. (2023) the potential of GPT (Generative Pre-trained Transformer) technology, particularly the use of Retrieval-Augmented Generation (RAG), to

advance clinical decision-making within healthcare. It underscores the significance of AI chatbots, powered by natural language processing and large language models like GPT, in engaging clinicians and patients in human-like conversations regarding health conditions and risks. These chatbots, while valuable, currently offer generic responses and lack the ability to provide individualized diagnostic and clinical guidance. The proposed solution involves integrating RAG into these AI chatbots, which can enhance the specificity of user prompts, enabling more detailed and contextually relevant responses. Furthermore, by incorporating current clinical data and reliable medical sources, such as clinical guidelines, into chatbot models, they may provide patient-specific assistance, speed diagnosis, and provide treatment suggestions, eventually leading to improved patient outcomes. The letter acknowledges the vast medical knowledge accessible to AI chatbots, even though they were not explicitly designed for healthcare, and suggests that RAG can harness this knowledge to deliver tailored and informed support, effectively bridging the gap between AI capabilities and personalized patient care. In summary, GPT technology, combined with RAG, holds the promise of significantly improving clinical decision-making and patient care by empowering AI chatbots to offer more precise and individualized responses to medical inquiries, thereby enhancing the healthcare experience for both clinicians and patients.

The paper by Chowdhury et al. (2022) introduces Retrieval Augmented Prompt Tuning (RAPT) as a parameter-efficient method to enhance paraphrase generation using large pre-trained language models. It also presents Novelty Conditioned RAPT (NC-RAPT) as a model-agnostic approach for controlled paraphrase generation with varying levels of lexical novelty. The study conducts extensive experiments on four datasets, demonstrating the effectiveness of these approaches in retaining the semantic content of the original text while inducing lexical novelty in the generated paraphrases. Evaluation metrics include BLEU for similarity, Self-BLEU and Self-TER for novelty, and BERT for semantic fidelity. RAPT

outperforms baselines in most aspects, with NC-RAPT allowing users to control trade-offs between novelty and other metrics. The study also highlights consistent results with a larger GPT-2 model. Future research may explore these methods with different language models and apply them to other downstream tasks like natural language inference and named entity recognition. Additionally, testing the effectiveness of the generated paraphrases for data augmentation is suggested as a potential avenue for further investigation.

In the paper by Bai et al. (2023) Mingliang Bai and his team describe an exceptional Large Language Model (LLM) made specifically for the renewable energy domain. The huge Renewable Energy Academic Paper (REAP) dataset, obtained from Web of Science, was used as the basis for data collecting, preprocessing, and fine-tuning an existing LLM. The foundation of their research is a dataset made up of more than a million scholarly articles. The Gradio library was also integrated by the authors into Python to produce a user-friendly web demo that enables users to use the model for content creation. Notably, their ground-breaking evaluation methods include both ChatGPT-based automatic assessments and human expert-guided Analytical Hierarchy Process (AHP) evaluations, enabling a comprehensive assessment of the model's effectiveness in producing academic articles specifically catered to renewable energy research.

By stressing the authors' techniques and technologies, highlighting the value of domain-specific LLMs, and introducing cutting-edge methods for LLM evaluation, this technical study highlights the authors' contributions. They create a strong framework for developing specialized LLMs tailored to particular domains through thorough data processing, strategic fine-tuning, and expert assessments, considerably boosting their utility in actual applications, particularly in the field of renewable energy research.

A study by Jeong (2023) offers a unique generative AI service implementation framework that successfully utilizes the power of LLMs, with a focus on the Retrieval-

Augmented Generation (RAG) model and LangChain orchestration. This platform, which incorporates a wide range of cutting-edge technology, makes it easier to incorporate generative AI into enterprises' everyday operations. Gathering source data, which comprises of documents in different file types such plain text, Microsoft Office documents, and PDFs, is the first step in the journey. These texts are meticulously broken down into smaller "chunks" using LangChain to improve performance. To facilitate the retrieval of critical information, the embeddings of these chunks are painstakingly created using OpenAI's robust models, or alternatively, GPT4All for web-based content. These embeddings are then meticulously maintained within a vector database, using ChromaDB or FAISS depending on the specific usage requirements, to enable speedy and accurate data retrieval. These embeddings allow the system to automatically pull pertinent portions from the database when a user submits a query. The context for large language models (LLMs), such as GPT-3.5-turbo or GPT4All from OpenAI, is then provided by these chunks. These LLMs deftly craft responses that are suitable for the situation. The paper's example implementation instances clearly show how these revolutionary technologies may coexist without conflict, demonstrating the great potential of AI-powered support chatbots to quickly and accurately respond to user inquiries, whether they are based on internal documents or online data.

In their paper Bonner et al. (2023b) show how Large Language Models (LLMs) can be used in the classroom to teach and learn languages. In this implementation, natural language creation and comprehension are based on the OpenAI GPT-3 paradigm. The authors provide examples of the numerous language-related tasks that LLMs, such as ChatGPT, can carry out, including text summarization, grammar checking, prompt generating, note-making, lesson idea development, and text leveling.

This implementation strategy focuses on designing specialized input prompts for ChatGPT, which specializes at comprehending and processing instructions in human-like

language. The model generates accurate responses to achieve a variety of goals, including text condensing, grammar checking, prompt development, and the generation of instructional concepts. For example, in text leveling, the authors provide the model instructions to produce texts appropriate for different language ability levels, such as CEFR B2. The study also highlights how flexible ChatGPT is in automating the creation of educational materials, including comprehension exercises based on given text. Overall, this study provides insightful information about the practical application of LLM technology in education, demonstrating the smooth integration of ChatGPT into language learning activities and giving teachers useful tools to improve their instructional strategies.

In research study by Chang et al. (2023b), the focus was on enhancing the capability of LLMs, such as GPT 3.5 and GPT-4, to generate text that is contextually relevant and useful in specific conditions, like having conversations with users. They achieved this by leveraging reinforcement learning (RL) techniques, which are like teaching the models to make better decisions as they learn.

What's unique here is that they used another LLM, like GPT-3, as a "guide" to help improve the main LLM's text generation. This guide LLM helped the main model learn how to produce better text. They also developed new RL algorithms, called RL with guided feedback (RLGF), tailored to the task of fine-tuning LLMs. To put their methods to the test, they applied them to various text generation tasks, such as creating positive movie reviews and other similar tasks. The results were quite promising. Their RL algorithms outperformed traditional methods like supervised learning (where the model learns from human examples) and the commonly used PPO (Proximal Policy Optimization) algorithm in terms of generating more accurate and contextually appropriate text. One noteworthy finding was that their approach even allowed them to learn from a very powerful model like GPT-3. In fact, their method using a simpler model (GPT-2) performed as well as GPT-3 in some cases. This

research opens up possibilities for improving text generation in various applications by combining RL techniques with the knowledge of powerful LLMs

Table 2

Technical Survey

Author	Domain	Key Technologies	Applications	Evaluation Metrics	Results/Performance	Future Directions
Cheng et al. (2023)	Adaptation of LLMs	LLM, domain adaptation	Various domains	Task-specific metrics	Improved LLM performance	Further exploration in domain adaptation
De Curtò et al. (2023)	Semantic Scene Understanding	LLMs, VLMs, object detection	UAV scene understanding	Performance assessment	Zero-shot semantic scene understanding	Integration, optimization
Siriwardhana et al. (2023)	Domain Adaptation in ODQA	RAG model, statement reconstruction	Open-Domain QA	ODQA metrics	Enhanced ODQA in specialized domains ²¹³	Trajectory optimization, additional sensor integration
Wang et al. (2023)	Clinical Decision-Making	GPT technology, RAG	Clinical decision-making	Healthcare metrics	Improved clinical decision-making	Real-time analysis, domain-specific tailoring
Chowdhury et al. (2022)	Paraphrase Generation	Paraphrase generation	Paraphrase generation	BLEU, Self-BLEU, BERT	Enhanced paraphrase generation	Application to other tasks
Bai et al. (2023)	Renewable Energy	LLM, LangChain	Renewable energy	Analytical Hierarchy	Domain-specific LLM	Optimization, domain-specific tailoring
Jeong (2023)	Generative AI Services	LLM, LangChain	Enterprise data	Pipeline performance	AI chatbot performance	Data augmentation testing
Bonner et al. (2023b)	Educational Tasks	ChatGPT, GPT-4, GPT-3.5 Turbo	Educational tasks	Educational metrics	Enhanced language learning	N/A
Chang et al. (2023b)	LLM Enhancement using RL	LLMs, RL	Text generation tasks	Text generation metrics	Improved text generation	Integration of RL in various applications
Jiang et al. (2023)	Natural Language Processing (NLP)	Grouped-Query Attention (GQA), Sliding Window Attention (SWA)	Reasoning, Mathematics, Code Generation	Inference Speed, Computational Costs	Outperforms Llama 1 (34B) and Llama 2 (13B)	Further optimization for real-time applications, exploration of additional applications, scalability improvements

Building upon recent advancements in Natural Language Processing (NLP), Jiang et al. (2023) introduce the Mistral 7B model, a 7-billion-parameter language model that marks a

significant stride in efficiency and performance. This model stands out by outperforming larger models like Llama 1 (34B) and Llama 2 (13B) in diverse tasks such as reasoning, mathematics, and code generation.

The key innovations of Mistral 7B, as outlined by Jiang et al. (2023), include the use of Grouped-Query Attention (GQA) and Sliding Window Attention (SWA). These mechanisms significantly accelerate inference speeds and effectively manage longer sequences, reducing computational costs and making Mistral 7B viable for real-time applications. The model's architecture allows it to outperform models with larger parameter counts, showcasing a balance between high performance and computational efficiency.

Jiang et al. (2023) also highlight Mistral 7B's fine-tuning capabilities, which allow it to adapt to various tasks, including the development of a chat model that outperforms similar class models. An essential feature for front-facing applications is the implementation of system prompts in Mistral 7B to enforce ethical guardrails, ensuring safe, respectful, and appropriate interactions with users.

Furthermore, Mistral 7B can perform content moderation with self-reflection, classifying prompts and responses as acceptable or unacceptable, a crucial feature for maintaining quality and safety in user-generated content platforms. In benchmark evaluations, Mistral 7B consistently outperforms existing models, especially in tasks involving code, mathematics, and reasoning.

1.5 Literature Survey

According to Guo et al. (2022) in the realm of natural language processing and healthcare, a significant research endeavor has emerged focusing on the development of advanced medical question answering systems. This particular study represents a pivotal contribution in this domain by amalgamating two cutting-edge technologies: large language models and knowledge graphs. Within the sphere of large language models, pretrained

models like BERT or GPT have garnered substantial attention for their remarkable language understanding capabilities. These models serve as the foundation upon which the medical question answering system is constructed, owing to their ability to comprehend context, semantics, and nuances in natural language. In parallel, knowledge graphs, which are structured representations of medical information, provide the structured domain-specific knowledge required to answer complex medical queries. This fusion of language models and knowledge graphs constitutes the crux of the innovation in this research, offering a promising solution for accurate and contextually rich medical question answering. Moreover, fine-tuning on medical domain-specific data is a technical aspect that enhances the system's ability to cater to the intricacies of the medical domain.

The paper Darapaneni et al. (2021) dynamic landscape of news and information dissemination has brought about a pressing need for sophisticated question-answering systems tailored to the intricacies of the news domain. In the context of natural language processing and information retrieval, this research paper marks a significant stride in addressing this requirement. It employs a multifaceted approach incorporating advanced techniques such as named entity recognition (NER), topic modeling, and document retrieval algorithms. NER plays a pivotal role in identifying entities of interest, such as individuals, organizations, and locations, within news articles. Simultaneously, topic modeling serves as a fundamental mechanism for deciphering the overarching themes and subjects within these articles. Furthermore, document retrieval algorithms are employed to efficiently locate and extract pertinent information. This confluence of technical components ensures the system's adeptness in navigating the ever-evolving news landscape, offering an effective means of accessing and comprehending information from diverse news sources.

The paper by Liu and Lapata (2019) says that art of text summarization within natural language processing has witnessed remarkable advancements, and this research paper

contributes significantly to this domain. Text summarization, often pivotal for condensing extensive textual content into concise and informative summaries, relies on state-of-the-art technologies such as pretrained encoder models. These encoder models, exemplified by BERT and Transformer-based architectures, have revolutionized language understanding and generation. The technical intricacies within this paper involve fine-tuning these pretrained models on summarization datasets, thereby imbuing them with the capacity to discern salient information and generate abstractive summaries. Additionally, decoding strategies like beam search or diverse decoding mechanisms are employed to optimize the generation process. Furthermore, the paper likely addresses challenges related to managing coherence in summaries, especially in the context of lengthy documents. This amalgamation of technical nuances results in the production of high-quality abstractive summaries, enhancing information accessibility and comprehension.

The paper by See et al. (2017) mentions that Within the realm of text summarization, the emergence of pointer-generator networks has introduced a novel dimension. These networks, a specialized category of sequence-to-sequence models, endeavor to bridge the gap between extractive and abstractive summarization techniques. In the context of natural language processing, this research paper makes noteworthy contributions by meticulously addressing the technical intricacies underlying pointer-generator networks. The crux of this innovation lies in the development of a hybrid loss function that encourages the model to make judicious decisions. It not only promotes the copying of words from the source document but also facilitates the generation of novel phrases. In addition to this, the system incorporates coverage mechanisms, a technical refinement aimed at ensuring that critical information is neither overlooked nor repeated during summarization. The outcome of these technical components is the generation of informative and fluent summaries, with applications spanning news, research, and content curation.

According to Manathunga and Illangasekara (2023) the synthesis of natural language processing and medical education presents a unique challenge - efficiently summarizing voluminous unstructured medical text. In response to this challenge, this research paper introduces a set of technical innovations. It delves into the realm of representation learning and summarization techniques. Document embeddings, generated using methods like Word2Vec or Doc2Vec, play a central role in encoding the essence of medical texts. The summarization process, in this context, often revolves around extractive techniques, wherein sentences or paragraphs are selected based on their significance. Furthermore, given the prevalence of medical terminologies, abbreviations, and domain-specific jargon, the paper likely addresses the technical complexities of ensuring that the summarization process remains precise and contextually relevant. These technical nuances culminate in a system capable of efficiently disseminating essential medical knowledge to facilitate accelerated learning and informed decision-making within the medical education landscape.

The paper by Krishna (2023) mentions intersection of question-answering systems and domain adaptation is a pivotal area within natural language processing, and this research paper makes significant strides in addressing this junction. It introduces the concept of few-shot domain adaptation, a technical innovation with far-reaching implications for open book question-answering systems. These systems, reliant on large language models, encounter challenges when confronted with new domains. The technical core of this paper revolves around adapting retrieval augmented generation models to excel in these scenarios. This adaptation process is facilitated through fine-tuning pretrained language models, such as GPT or T5, on limited domain-specific data. Additionally, prompt engineering and transfer learning techniques likely play pivotal roles in enhancing the models' adaptability to new domains, even when minimal training samples are available. Consequently, this research

contributes to the development of versatile question-answering systems capable of providing accurate responses across diverse subject matters.

In the realm of large language model applications according to Topsakal and Akıncı (2023) the expedited development of robust and efficient applications has emerged as a focal point of interest. This paper introduces LangChain, a comprehensive framework that streamlines the development of applications based on large language models. This framework, within the context of natural language processing and software engineering, encompasses several technical components. It likely incorporates containerization for efficient deployment, enabling seamless scaling and distribution of applications. Furthermore, the incorporation of continuous integration/continuous deployment (CI/CD) pipelines ensures a systematic and agile approach to application development. The modular architecture design is another technical facet that contributes to the framework's flexibility, allowing for the integration of various use cases and components. Furthermore, LangChain is anticipated to provide user-friendly APIs that facilitate developers' interactions with language models, simplifying the integration of these models into diverse applications. This comprehensive approach holds the potential to revolutionize the landscape of language model-based applications by significantly reducing development time and enhancing accessibility.

According to Pavlyshenko (2023) the department of financial analysis is intrinsically tied to the processing and interpretation of financial news, and this research paper brings to the fore the fusion of language models and financial data analytics. Within the domain of natural language processing and financial analysis, technical nuances encompass the fine-tuning of a pretrained language model, possibly GPT or a similar architecture, on specialized financial news datasets. The incorporation of financial lexicons and domain-specific embeddings is a crucial technical aspect, enhancing the model's grasp of financial terminology and context. Moreover, this paper likely delves into techniques such as sentiment

analysis, event detection, and entity recognition, all pivotal for extracting valuable insights from financial news articles. These technical components contribute to the paper's overarching goal - aiding investors and analysts in making well-informed financial decisions and comprehending market trends through the lens of large language models.

The integration of knowledge governance within the telecommunications sector, given in paper by Cai and Wu (2023) mention about how this sector underpinned by large language model applications, stands as a significant focal point in this research paper. Within the scope of natural language processing and telecom knowledge management, this paper introduces a comprehensive knowledge governance framework. This framework, inherently technical in nature, encompasses several pivotal aspects. Data pipelines likely play a prominent role, facilitating the collection of telecom-specific data from diverse sources. Integration with large language models and associated techniques for knowledge representation and retrieval is central to the framework's functioning. Security and privacy measures also occupy a significant place, given the sensitivity of telecom data. Furthermore, the framework may incorporate advanced data analytics and visualization tools, offering a holistic approach to extracting actionable insights from textual data specific to the telecom industry. This technical framework represents a pivotal contribution, ensuring that large language model applications within the telecommunications sector are well-equipped to meet industry-specific requirements and challenges.

In their pivotal work, Jiang et al. (2023) introduce Mistral 7B, a language model that innovatively balances high performance with computational efficiency. With 7 billion parameters, Mistral 7B stands out in the Natural Language Processing (NLP) landscape, challenging traditional computational cost norms and scaling models. The essence of this model lies in its unique architecture and attention mechanisms, namely grouped-query attention (GQA) and sliding window attention (SWA). These features not only enhance

inference speed but also adeptly manage longer sequences, substantially reducing computational overhead.

Table 3

Literature Survey

Authors	Domain/Application Focus	Key Technologies	Contributions
Guo et al. (2022)	Healthcare	Large Language Models, Knowledge Graphs	Integration of language models and knowledge graphs for accurate medical question answering; fine-tuning on medical data.
Darapaneni et al. (2021)	News	NER, Topic Modeling, Document Retrieval	Multifaceted approach for news domain QA; entity recognition, topic modeling, and document retrieval.
Liu and Lapata (2019)	Text Summarization	Pretrained Encoder Models, Summarization	Fine-tuning of encoder models for abstractive summarization; optimization strategies; coherence management.
See et al. (2017)	Text Summarization	Pointer-Generator Networks	Hybrid loss function, coverage mechanisms for informative and fluent summaries; bridging extractive and abstractive methods.
Manathunga and Illangasekara (2023)	Medical Education	Document Embeddings, Extractive Summarization	Representation learning, extractive summarization for medical education; addressing medical jargon and precision.
Krishna (2023)	Question-Answering, Domain Adaptation	Retrieval Augmented Generation Models	Few-shot domain adaptation for QA systems; fine-tuning and prompt engineering for versatile domain adaptation.
Topsakal and Akinci (2023)	Large Language Model Applications	LangChain Framework	Framework for efficient LLM application development with containerization, CI/CD, modular design, user-friendly APIs.
Pavlyshenko (2023)	Financial Analysis	Fine-Tuned Language Models, Sentiment Analysis	Fine-tuning on financial datasets; sentiment analysis, event detection, entity recognition for financial insights.
Cai and Wu (2023)	Telecom Knowledge Management	Knowledge Governance Framework	Data pipelines, knowledge representation, security measures, analytics for telecom knowledge governance.
Jiang et al. (2023)	Natural Language Processing (NLP)	Grouped-Query Attention (GQA), Sliding Window Attention (SWA)	Mistral 7B is a 7-billion-parameter language model in NLP, uniquely balancing high performance with computational efficiency through innovative architecture and attention mechanisms (GQA and SWA). Its transformative approach includes fine-tuning capabilities for specialized tasks, ethical guardrails for safe user interactions, and self-reflective content moderation, consistently outperforming in benchmark evaluations, especially in code, mathematics, and reasoning tasks

Jiang et al. (2023) also emphasize Mistral 7B's adaptability in being fine-tuned for various specialized tasks, including a chat model that outperforms similar models in its class.

The incorporation of system prompts to maintain ethical guardrails marks a significant step towards responsible AI deployment, ensuring safe user interactions. Moreover, the model's self-reflective content moderation capability, classifying prompts and responses for quality control in user-generated content, highlights its practical applicability.

Conclusively, Jiang et al. (2023) showcase Mistral 7B as a model that revolutionizes the approach to LLM development, combining technical sophistication with ethical considerations. Its ability to reduce development time and enhance accessibility positions it as a transformative force in the realm of language model-based applications, both in terms of affordability and efficiency.

In summary, each research paper, within the realm of natural language processing and large language model applications, contributes a wealth of technical innovations and insights. These innovations span diverse domains, from healthcare and news to text summarization, domain adaptation, financial analysis, and telecom knowledge governance. As these papers collectively advance the landscape of natural language processing, they address specific challenges while contributing valuable knowledge and methodologies to the broader field.

2. Data and Project Management Plan

2.1 Data Management Plan

Data Collection Approaches

ARXIV is a free and open-access online platform that acts as a preprint repository for academic and scientific research articles. This project's data comes from two distinct sources. The primary source is the arXiv database which contains more than 2 million articles and papers for research study. The beauty of this data source is, it is stored in the server which is maintained in a very organized manner and can be accessed directly from the ARXIV website itself and also it is easy to get huge number of pdf Documents at once to perform word processing and cleaning.

Although this site has vast majority of the articles, this project mainly concentrates on computer science research papers. In our data collection process, we focus on machine learning research papers within the ARXIV database. Machine learning is a constantly expanding area that creates a massive corpus of information on a continual basis. Our goal is to create a specialized dataset by extracting and obtaining data on the most recent advancements, approaches, and discoveries in the field of machine learning. This dataset becomes a crucial asset for our research and analysis endeavors. Our approach involves both extraction and downloading of relevant data from ARXIV. This entails systematically identifying and retrieving papers, articles, and associated metadata that pertain to machine learning algorithms.

The second form of data source used is the Hugging face/QASPER dataset. This HUGGING FACE is a community platform where all the algorithms starting from basic ML Algorithms to large LLMS are embedded into it along with datasets so that those datasets can be used which is more aligned with the contextual based on reading and understanding the system.

In terms of this data source, QASPER is a specialized dataset designed for the purpose of question answering within the domain of scientific research papers, with a particular focus on Natural Language Processing (NLP) literature. This dataset comprises a total of 5,049 questions, which are meticulously crafted by NLP practitioners. These practitioners, however, are restricted to read only the title and abstract of the respective research papers.

The unique aspect of QASPER lies in the nature of these questions. They are skilfully formulated to seek information that is exclusively present in the full text of the research papers, extending beyond what can be gleaned from just the title and abstract. This approach mimics a real-world scenario where a user may have access to limited information about a document but needs to extract specific details from its complete content.

Figure 1

Sample QASPER JSON raw DATASET to be included for Fine Tuning Approach

```
{
  "id": "Paper ID (string)",
  "title": "Paper Title",
  "abstract": "paper abstract ...",
  "full_text": {
    "paragraphs": [{"section1_paragraph1_text": "section1_paragraph1_text", "section1_paragraph2_text": "section1_paragraph2_text", "section_name": "section1_title"}, {"section2_paragraph1_text": "section2_paragraph1_text", "section2_paragraph2_text": "section2_paragraph2_text", "section_name": "section2_title"}],
    "qas": [
      {
        "answers": [
          {
            "annotation_id": ["q1_answer1_annotation_id", "q1_answer2_annotation_id"],
            "answer": [
              {
                "unanswerable": false,
                "extractive_spans": ["q1_answer1_extractive_span1", "q1_answer1_extractive_span2"],
                "yes_no": false,
                "free_form_answer": "q1_answer1",
                "evidence": ["q1_answer1_evidence1", "q1_answer1_evidence2", "..."],
                "highlighted_evidence": ["q1_answer1_highlighted_evidence1", "q1_answer1_highlighted_evidence2", "..."]
              },
              {
                "unanswerable": false,
                "extractive_spans": ["q1_answer2_extractive_span1", "q1_answer2_extractive_span2"],
                "yes_no": false,
                "free_form_answer": "q1_answer2",
                "evidence": ["q1_answer2_evidence1", "q1_answer2_evidence2", "..."],
                "highlighted_evidence": ["q1_answer2_highlighted_evidence1", "q1_answer2_highlighted_evidence2", "..."]
              }
            ],
            "worker_id": ["q1_answer1_worker_id", "q1_answer2_worker_id"]
          }, ...
        ],
        "question": ["question1", "question2", "question3", "..."]
      },
      {
        "question_id": ["question1_id", "question2_id", "question3_id", "..."],
        "question_writer": ["question1_writer_id", "question2_writer_id", "question3_writer_id", "..."],
        "nlp_background": ["question1_writer_nlp_background", "question2_writer_nlp_background", "question3_writer_nlp_background", "..."],
        "topic_background": ["question1_writer_topic_background", "question2_writer_topic_background", "question3_writer_topic_background", "..."],
        "paper_read": ["question1_writer_paper_read_status", "question2_writer_paper_read_status", "question3_writer_paper_read_status", "..."],
        "search_query": ["question1_search_query", "question2_search_query", "question3_search_query", "..."]
      }
    ]
  }
}
```

Figure 2

Sample raw DATASET to be included for first Approach.

Name	Owner	Last modified	File size	⋮
2310.05921v1.pdf	me	9:30PM me	4.3 MB	⋮
2310.05920v1.pdf	me	9:30PM me	18.4 MB	⋮
2310.05918v1.pdf	me	9:30PM me	3.7 MB	⋮
2310.05915v1.pdf	me	9:30PM me	767 KB	⋮
2310.05914v1.pdf	me	9:30PM me	783 KB	⋮
2310.05910v1.pdf	me	9:30PM me	939 KB	⋮
2310.05905v1.pdf	me	9:30PM me	1.3 MB	⋮
2310.05900v1.pdf	me	9:30PM me	1.9 MB	⋮
2310.05898v1.pdf	me	9:31PM me	1MB	⋮
2310.05897v1.pdf	me	9:31PM me	2.7 MB	⋮
2310.05893v1.pdf	me	9:31PM me	721 KB	⋮

Data Storage and Management Methods

In terms of data storage, we used two different storage mechanism for different data sources. The various collected papers are stored in the vector database which is used for vector mapping to fetch the relevant documents based on the user query. Most of the vector databases are cloud supported platform which can be accessible 24/7. The data will be stored in two regions for any disaster recovery in one of the region.

In our data storage strategy, we've adopted a tailored approach, utilizing two distinct storage solutions to accommodate the unique needs of different data sources. For the vast collection of research papers gathered, we've opted for a vector database as the storage medium of choice. This vector database serves a pivotal role in our research endeavors by employing vector mapping techniques to efficiently retrieve relevant documents based on user queries. Notably, these vector databases are hosted on cloud-supported platforms, ensuring accessibility around the clock, 24/7. This accessibility is vital for researchers and users who may need to access or query the stored papers at any given time.

To further fortify our data storage strategy and enhance disaster recovery capabilities, we've implemented a redundancy plan. This involves replicating our data in two separate regions, thereby providing resilience and safeguarding against potential data loss in the event of a disaster affecting one of the regions. This disaster recovery setup ensures the continuity and integrity of our research data, assuring that it remains available and secure even under adverse circumstances.

In essence, our structured data storage approach optimizes accessibility and reliability for the research papers, leveraging vector databases on cloud-supported platforms while maintaining data redundancy for enhanced disaster recovery measures.

The second data source which is QASPER and hugging face is stored in the google cloud platform in a very organized manner. As we know the machine learning papers will be spiked up for the years of advancement in the technology and more number of papers will be evolved and datasets will be evolved as well simultaneously. Currently the ZIP folder we extracted is around 4 GB in size which in turn will be 12 GB up on the extraction.

To preserve this scalability, we've allotted 25GB of storage capacity across the two zones. The dataset's first copy is in Los Angeles, California. The second copy is kept in Dallas, Texas, and will be utilized for data availability as well as catastrophe recovery. The python script has been written in such a way that data is collected from the QASPER/Hugging face framework and processed data is put in the database.

Table 4

Structure to be implemented in GCP

Purpose	Folder Name	File Name
RAG Approach	Research_Papers	2022_papers
		2023_papers
Finetuning Approach	QASPER	questionnaire.csv

There is set up for roles that control who can do what in our project using IAM policies. Each team member has specific permissions based on their job. If someone needs more access or changes to what they can do, they have to ask the administrator first, and only after getting approval will they get access.

Data Usage and Mechanisms

The research paper dataset's accessibility is a primary focus. The source code is accessible and will be available on GitHub. The website's dataset is continuously updated, with new columns added on a monthly basis. The GitHub Repository contains detailed project documentation, including the dataset and code explanations.

During the EDA process, we took efforts to strengthen data security. The revised dataset may be found on GitHub over a secure SSH connection. To save Google Cloud Storage expenses, data on GCP will be erased once all submissions have been completed. This includes datasets stored throughout the year, which lowers resource allocation costs and is readily accomplished by removing the associated buckets on the Google Cloud Platform.

2.2 Project Development Methodology

This project is adopting the CRISP-DM methodology for its development, and Table 5 provides a summary of the assigned tasks for each team member.

Business Understanding

In this project, we're tackling a significant challenge related to the sheer number of academic papers being published in the field of Machine Learning (ML). Imagine this: every single day, about 100 new ML papers are published worldwide, adding up to a whopping 36,500 papers every year. This flood of research publications has created a real problem – it's overwhelming for researchers, students, and professionals who want to keep up with the latest developments in ML. The central issue here is finding a way to make it easier and faster

for people to access, understand, and engage with these ML research papers, essentially making knowledge sharing in the field more efficient.

The objectives of this project are multi-faceted and carefully delineated to address the complex issue at hand. Firstly, the main aim is to optimize data storage and retrieval by seamlessly integrating ML research papers from the arXiv database into the Pinecone vector database. This integration is poised to significantly enhance the efficiency of search and analysis. Secondly, the task is to develop a Large Language Model (LLM) that leverages cutting-edge techniques such as Llama 2, Mistral 7B and retrieval augmented generation. The LLM's primary mission is to bolster text comprehension and generation, equipping it with the proficiency to understand and generate precise responses pertaining to ML research. Thirdly, the LangChain's abstract concepts are employed, encompassing Agents for task handling, Memory for learning, and Chains for sequences. These abstractions play a pivotal role in assisting users in comprehending intricate ML papers, furnishing contextually relevant responses, and enabling deeper insights into technical content. The project also places strong emphasis on evaluating the LLM's performance rigorously, deploying measures like EVALS, perplexity, BLEU score, ROUGE score, and fidelity to ensure that it consistently delivers coherent and contextually pertinent responses. Lastly, a user engagement system is developed by introducing a questionnaire interface, empowering users to seek clarification and gain profound insights into the technical nuances of ML papers, thereby creating a more interactive and user-centric platform.

Prior to embarking on this, comprehensive research was conducted into existing reading and understanding systems within the realms of ML and Natural Language Processing (NLP). This preliminary exploration aimed to shed light on the strengths, weaknesses, and gaps present in current solutions. By examining the landscape of existing systems, major invaluable insights are taken into consideration on what works effectively and

identified persistent challenges. This foundational research served as a crucial precursor to shaping our project's direction and ensuring that it would contribute meaningfully to the evolving field of ML research paper comprehension and accessibility.

Table 5

Responsibility and Resource Allocation for DMP

DMP Phase	Resource Allocation	Work Assigned
Data Collection	Kalyan	Python script is developed to gather the convert parquet files to CSV files using Apache spark for QASPER data
	Dharnidhar	Retrieval of PDFs from arXiv database for Machine Learning Research Papers
Documentation and Metadata	Manideepya	Documented the details of methodology used and assumptions considered on this data
Ethics and legal compliance	Nikhil	All softwares used in this project are been taken from the official websites with registration of proper licensed keys in order by agreeing to their terms and conditions.
Storage and Backup	Dharnidhar	Data is stored in the Google Cloud Platform and for recovery data is replicated across multiple locations before and after embedding
Selection and preservation	Manideepya	Data which is frequently used and with the highest importance should be preserved in multiple locations
Data Sharing	Kalyan	All the data and code will be uploaded on to github repository with appropriate permissions

Data Understanding

In the data understanding phase the first step is to have a plan for collecting and analyzing machine learning research paper and then to categorize the types of different ML papers that are needed. Additionally, to establish a clear timeframe for data collection process is done which involves deciding whether the data should have papers from a specific range of years or maintain an ongoing dataset, continually updated with new publications.

With the data needs being comprehensively outlined, the next phase will be the data collection phase. The primary source for research papers is the renowned arXiv database. To achieve this, a robust data pipeline capable of fetching papers from arXiv will be established. Leveraging arXiv's API or employing web scraping techniques, we ensured efficient and lawful data retrieval. Simultaneously, another dataset from huggingface which is provided by "QASPER" is taken for the subsequent development of the data collection pipeline.

Once data collection was underway, we confronted the challenge of extracting valuable information from PDF documents—a common format for research papers. Employing specialized PDF parsing libraries such as PYPDF2, which extracts not only the complete text of each paper but also crucial metadata. This metadata encompassed vital details like paper titles, author lists, abstracts, publication dates, and keywords. It's important to note that our approach had to account for different variations in PDF layouts and structures, as different sources might present papers differently.

Ensuring data quality is also an important phase in this project, thus, there will be a substantial effort for cleaning the data. To avoid redundancy and enhance data accuracy, the de-duplication techniques will be implemented. Duplicate papers will be identified and subsequently removed from the dataset. These duplicates could originate from multiple versions or submissions of the same paper. To achieve this, techniques such as fuzzy matching and the utilization of unique identifiers like Digital Object Identifiers (DOIs) will be implemented.

After obtaining this cleaned and structured data from the unstructured data, the next process involves assessing its quality. Which consists of a thorough analysis of the extracted metadata and textual content where the missing data, inaccuracies, and inconsistencies will be checked, resulting in ensuring that the obtained dataset is reliable and complete. Additional

validation is conducted by cross-referencing metadata against external sources or conforming to established formats to verify the accuracy of the obtained dataset.

The final phase of the Data Understanding process consists of Exploratory Data Analysis (EDA). This step helps to gain deep insights of the datasets. For the research papers dataset from the arxiv database, an in-depth analysis of publication trends over time, the distribution of paper topics, authorship patterns, and more have been conducted. Meanwhile, for the QASPER dataset which is obtained from Hugging Face the structure, content, and any potential relationships with the research papers data are analyzed. The analysis is done using various visualization tools and techniques. This thorough understanding of the data serves as a robust foundation for subsequent stages in this project, ensuring that the data is efficient to create a user-friendly interface for analyzing machine learning research papers effectively.

Data Preparation

The Data Preparation phase is a crucial part that ensures the datasets are well-structured and ready for the implementation of the LLM model. To obtain this, the initial steps involve converting the textual content of research papers, stored in PDF format, into manageable and structured chunks. To accomplish this task, the PyPDF2 library is used, which helps in extracting the text from PDF documents. Each chunk represents a cohesive part of the PDF's which helps in understanding and retrieving the data.

The main approach of this project is the conversion of the obtained chunks into embeddings utilizing word embedding models. These embeddings capture both the semantic and contextual information within each chunk, providing a comprehensive representation of the text. This will enhance the dataset's richness and enables the Large Language Model (LLM) to better comprehend and generate text.

The next step involves the seamless integration of the research papers dataset which are in the form of embeddings, into the Pinecone vector database. Pinecone's vector indexing

and retrieval capabilities ensure quicker and more effective searches and analyses which guarantees that the user interface will provide users with rapid access to relevant research papers, offering a seamless and efficient user experience.

In parallel with our PDF to chunk conversion and embedding process, the questionnaire data from the QASPER dataset is extracted. This data holds the valuable information regarding the user preferences, interests, and areas of curiosity with respect to the machine learning research papers. Incorporating this questionnaire data into this analysis, the aim is to develop a better interface which serves the users by addressing their specific needs and queries.

For Large language model implementation and evaluation, the questionnaire dataset is partitioned which now include embedded training and testing subsets. This helps in estimating the model's performance by rigorous assessment, which leads to preventing overfitting and ensuring generalizability. The training data will be used to train the LLM, while the testing data will be utilized to evaluate their effectiveness.

The final stage of data preparation for this project relies on the successful environment setup having a well-configured software and hardware environment. An environment that accommodates the diverse set of dependencies will be required for this project, including software packages, frameworks and install required libraries. Selection of an appropriate pre-trained LLM model and allocate necessary hardware resources and implementation of tokenization for fine-tuning which is capable of handling the computational demands of the project, guaranteeing efficient data processing and model training.

Modeling

In the Modeling Phase of this project involves the process of implementing the capabilities of a Large Language Model (LLMs) to enhance the understanding and generation

of text from machine learning research papers. This is done by initializing LLM models, specifically Llama2 7B and 13B is incorporated for this project with pre-trained weights. These pre-trained models, developed with vast text corpora, serve as the foundation for building the reading and understanding system. Implementation of the models with pre-trained weights helps in leveraging the knowledge from diverse textual sources, enhancing the language understanding capabilities.

The training of the LLM is done using the questionnaire dataset which helps in achieving the outcome for this project based on the user requirements. The training process will involve exposing the models to the questionnaire data, allowing them to learn and internalize the patterns, semantics, and user preferences encoded within this dataset. This stage plays a crucial role in customizing LLMs to offer responses to user queries that are more contextually relevant.

Fine-tuning for this done by employing the state-of-the-art techniques, namely LoRA (Language Model Ranking) and QLoRA (Question-based Language Model Ranking) from the Pre-training Efficiency Framework for Transformers (PEFT). These techniques fine-tune the LLM models by leveraging ranking-based training, allowing the models to better rank and generate responses based on relevance to user queries. Fine-tuning not only refines the models but also significantly enhances their performance in delivering precise and coherent text.

To further enhance the models' capabilities the hyper-parameter tuning will be implemented. This optimization process involves adjusting various hyper-parameters, such as learning rates, batch sizes, and model architecture configurations. Hyper-parameter tuning seeks to strike the right balance between model performance and resource efficiency, ensuring that the LLMs will operate at their peak potential.

Evaluation

The Evaluation Phase of this project is done for assessing the effectiveness and user-friendliness of the ML papers reading and understanding system, which incorporates Large Language Models (LLMs) for enhanced comprehension and generation of textual content. This phase will involve a comprehensive evaluation to gauge the quality and coherence of the textual content generated by the LLM-based system. This evaluation phase will include both human evaluation and automated metrics like the ROGUE Score. Human evaluators will assess the system's responses for their relevance, coherence, and contextual accuracy. The ROGUE Score, on the other hand will provide quantitative insights into the system's performance in generating summaries or responses that align with human-generated references.

User testing will be done as the next step in evaluating the real-world usability and user-friendliness of this ML papers reading and understanding system. Real users interacted with the system, posing queries, seeking clarification, and exploring research papers. During this phase, the user feedback regarding their experience is considered as the more valuable insight for their ease of use, and the system's ability of providing the relevant information.

The user testing data will be rigorously analyzed to gain insights of the strengths and weaknesses of this system. Identifying areas where the system has excelled and pinpointing aspects that required improvement will be taken into consideration for further improvement. This analysis will be formed as the basis for iterative enhancements and refinements to ensure the system will align with user expectations and demands.

In this process the evaluation and validation will be done in an iterative process to maintain consistent improvement of the system. Based on the user feedback and the metrics obtained the instruction tuned LLM models, fine-tuning parameters, and system components will be re-evaluated. Each iteration will be aimed to enhance the system's performance, responsiveness, and adaptability based on the varying user needs and queries. This iterative

approach ensures that the system evolution will make sure to meet user expectations effectively.

Deployment

The initial step in the deployment phase for this project will be the integration of our ML papers reading and understanding system with the user interface. This critical integration will ensure that users can easily access and interact with the system through a user-friendly interface where the system's capabilities are readily available to users, enabling them to explore machine learning research papers with ease.

When the users begin to interact with this system through the integrated user interface, there will be a continuous valuable feedback collection process. This feedback will provide valuable insights into how users perceive and utilize the system and assess with a keen focus on identifying areas where the system excels and areas where further improvements or modifications are necessary. This iterative approach to improve the system based on the alignment with user expectations and demands will ensure in delivering a more refined and impactful experience.

Comprehensive documentation is a crucial aspect of the Deployment Phase. The documentation consists of the models used, including details about LLMs, embeddings, and other components which are implemented to build the reading and understanding system for the ML Research papers. Additionally, the findings from the Evaluation Phase, highlighting performance metrics, user testing results, and insights gained through user feedback analysis are also documented. This documentation serves as a valuable reference and knowledge that will help for future development and improvement which can be made for this system.

The entire code will be documented in a GitHub repository along with the instructions for setting up the environment for the implementation of the system, and for making contributions for future collaborations having transparency along with the capability of

reproductivity feature. By hosting the project on GitHub, the aim is to encourage open-source collaboration and ensure that the codebase remains accessible and well-maintained.

2.3 Project Organization Plan

The goal of the project is to create a user-friendly interface which will effectively analyze and understand the Machine Learning (ML) research papers. To achieve this, a structured and effective approach which is CRISP-DM methodology having six key phases is implemented by the project team.

Business Understanding is the initial phase where the team members will work on problem definition and objectives, and understand the specific requirements of the project and the needs of the target audience. The data Understanding is the second phase where the focus of the project team members will be on collecting a comprehensive set of research papers by exploring various data sources, and assessing the quality of the data which ensures that the system has access to a diverse and reliable dataset.

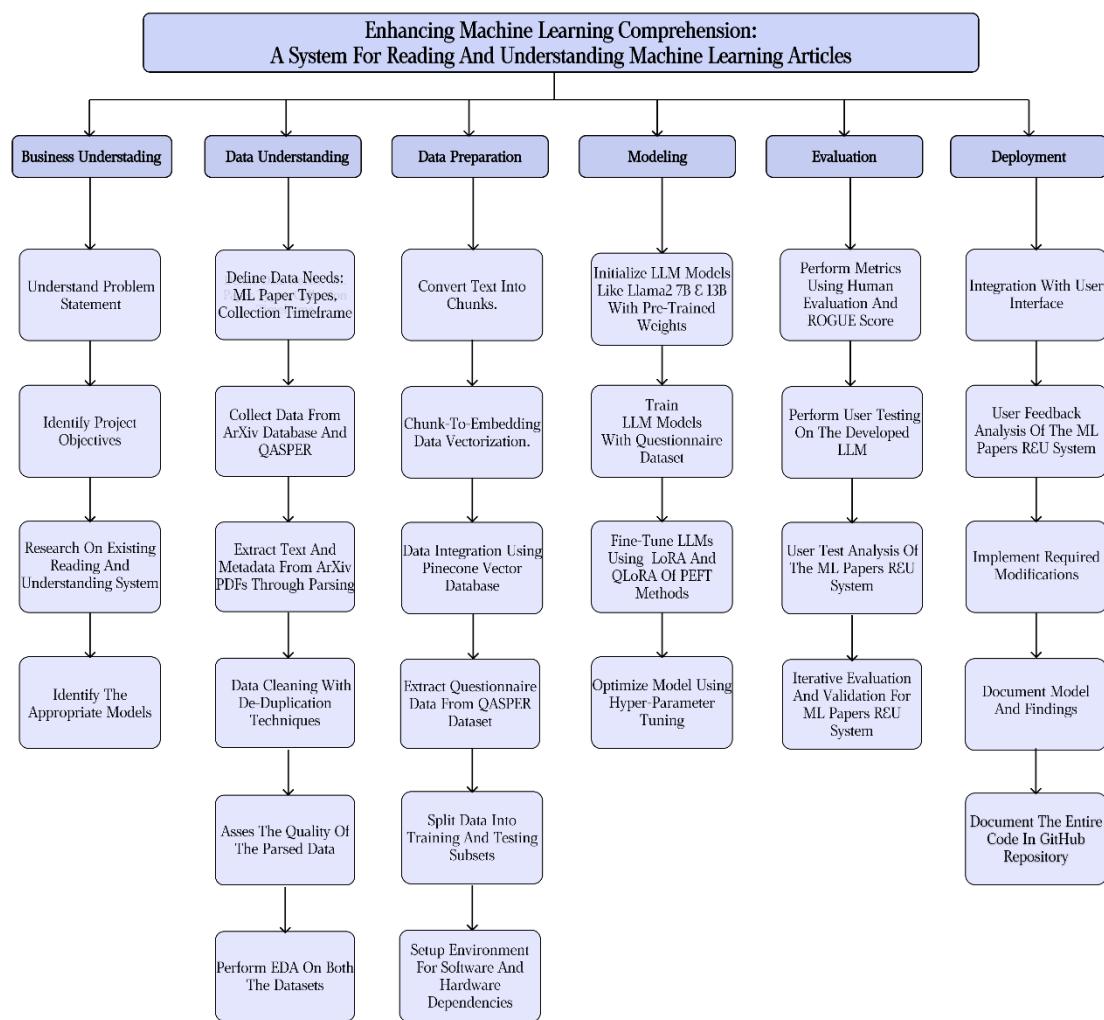
Data Preparation is the next phase where the data collected in the previous phase will be cleaned and integrated effectively by the team. Here, it is checked whether the input data is in an optimal format for implementing the modeling phase. And coming to the modeling phase it is where the heart of the project lies. In this phase, a Large Language Model (LLM) will be designed to enhance text comprehension and generation, making it the important component of the system.

In the Evaluation phase the team will measure the performance of the LLM and the overall system, various metrics such as ROUGE will be employed. Additionally, this will also involve gathering user feedback which will play a crucial role in refining the system based on real-world usage. Deployment will be the final phase that involves integrating the system with a user-friendly interface which will also involve Ongoing monitoring and maintenance analysis to ensure that the system remains effective and continues by meeting the user needs.

This UI will help to improve the accessibility and comprehension these complex research papers which helps to seek clarification, and gain deeper insights on the papers. By leveraging the capabilities of a Large Language Model (LLM) to enhance text comprehension and provide effective user assistance

Figure 3

Work Breakdown Structure for Reading and Understanding System for ML Papers



2.4 Project Resource Requirements and Plan

There is a requirement to collect a wide range of essential resources in order to assure the success of the "Reading Understanding System for Machine Learning Research Papers" project. These sources provide the framework, similar to the bricks needed to build a strong structure, that will

enable us to create a reliable and efficient tool for understanding complex machine learning research papers.

Hardware Requirements

There is need access to resources for high-performance computing first and foremost. These resources may come in the form of cloud-based servers or powerful machines located on-site. The Central Processing Unit (CPU) within these computing resources must be capable; this is of highest significance. The system's brain, the CPU manages a variety of duties quickly and effectively. This is particularly important when it comes to training the huge language models, which demand sophisticated data processing and mathematical calculations. This project also require graphics processing units, or GPUs, in addition to the CPU. As GPUs are specialized hardware components designed for parallel computing and are exceptionally well-suited for ML tasks.

Table 6

Hardware Requirements

Hardware Name	Configuration	Purpose
Local Machine	12 CPUs, 16 GPUs	To develop and build the models
Graphics Processing Unit (Google Colab GPUs)	16 GB High Bandwidth Memory	Specialized hardware components designed for parallel computing, which can drastically cut down the amount of time needed to train and use the system. The GPU will be used to train and use machine learning models to understand the meaning of the text and answer questions about it.
Random Access Memory (RAM)	16GB or more	Supplements the CPU and GPU's processing capabilities and acts as the system's short-term memory, enabling quick access to and manipulation of data. The RAM will be used to store the text of the articles, as well as the data used to train and use the machine learning models.

They drastically cut down the amount of time needed to fine-tune the system by executing several computations simultaneously to speed up the training of these models. A sufficient amount of RAM is necessary to supplement the CPU and GPUs' processing capabilities. The system's RAM acts as its short-term memory, enabling quick access to and manipulation of data. In order to ensure that data can be processed and manipulated effectively throughout the training and use of machine

learning models, especially LLM models, this high-speed memory is essential. These hardware requirements, taken as a whole, serve as the framework for this project. They enable us to quickly, accurately, and effectively read and comprehend machine learning research articles, which eventually results in the successful implementation of this Reading Understanding System for ML articles.

Software Requirements

Our project relies on a wide range of software tools and libraries, all of which help us realize the vision for the "Reading Understanding System for Machine Learning Research Papers." These digital elements work together in harmony to create the system's powerful capabilities.

Development Tools. At the forefront of this project software toolkit are development tools that provide the essential workspace for this project. The coding journey for this project starts on platforms like Jupyter Notebook and in integrated development environments (IDEs). These user-friendly interfaces make it easier to write, test, and improve the software code, allowing us to develop the project's technical ability.

Libraries. A wide variety of libraries, each with a special function, further enhance the software development journey. This project system's machinery is propelled by these libraries, which provide a wide range of functionality. They are crucial for the research, helping with everything from data processing and analysis to perfecting the machine learning models. Among them are Scikit-Learn, Numpy, Pandas, Hugging Face Transformers, and Langchain specific libraries like PyPdfLoader, LLMTainer,LLMDeployer and others that are important to the growth of this project.

Database Management. A reliable database system is a key component of this project. As the strong framework of this system, this digital repository may store enormous collections of academic publications, user interactions, and semantic embeddings. It serves as the custodian of the priceless data for this project, ensuring quick access and management

Machine Learning Frameworks. The power of machine learning frameworks, particularly TensorFlow or PyTorch, runs through the center of this system. The machine learning models are propelled by these frameworks, which enable us to precisely build, train, and optimize them.

Semantic Search Tools. The project employs specific tools or libraries like Pinecone Vector deliberately developed for this purpose in order to fully exploit semantic search's tremendous

potential. These techniques advance the system beyond the limitations of straightforward keyword-based searches and allow us to explore the complex contextual environment of research publications. They give us the ability to deliver users with search results that are both highly relevant and more meaningful.

Table 7*Software Requirements*

Category	Software Tool	Purpose
Development Tools	Jupyter Notebook	Interactive environment for developing and running Python code
Development Tools	Integrated development environments (IDEs)	PyCharm
Libraries	LangChain	Framework for building applications using large language models (LLMs)
Libraries	PyPDFLoader	Python library for loading and parsing PDF files
Libraries	Llama 2	Large language model (LLM) that can be used for a variety of tasks, including text generation, translation, and question answering
Libraries	Pinecone Vector	Vector store that can be used to store and retrieve vectors
Libraries	Hugging Face Transformers	Library that provides pre-trained and fine-tuned transformer models for a variety of tasks, including natural language processing (NLP), machine translation, and text generation
Libraries	Scikit-Learn	Python library for machine learning
Libraries	Numpy	Python library for scientific computing
Libraries	Pandas	Python library for data analysis
Database Management	PostgreSQL	Relational database management system (RDBMS)
Machine Learning Frameworks	TensorFlow	Open-source software library for numerical computation using data flow graphs
Machine Learning Frameworks	PyTorch	Open-source machine learning framework based on the Torch library
Semantic Search Tools	Pinecone Vector	Vector store that can be used to perform semantic search
Version Control	GitHub	Web-based hosting service for version control using Git

Version Control. The usage of GitHub allows for easy collaboration and code management.

For a virtual workspace it is made possible by this widely-used platform, which keeps this project

well-organized and accessible. It serves as the central location for team members to collaborate, monitor changes, and easily maintain version control.

Tools And Licenses

It's important to understand the license details of the tools and resources that will be used in the software landscape of this project. Here is a list of the licenses connected to the main parts of this project.

Llama 2. A key component of this project is the formidable large language model (LLM) called Llama 2, which was created by Meta. The good news is that Llama 2 is an open-source project and that it has a permissive license that permits a variety of uses, including commercial ones. It is simple to obtain the Llama 2 license; it is done by downloading the model from the Hugging Face website. Llama 2 also comes in a variety of sizes, so you may pick the one that best suits the needs of your project. Mistral 7B. Same as the Llama 2, even Mistral is an open source model.

Table 8

Tools And Licenses

Tool Name	License	Purpose
Llama 2	Free	Large language model (LLM) for text generation, translation, and question answering
Mistral 7B	Free	Large language model (LLM) for text generation, translation, and question answering
Pinecone Vector	Free (limited features)	Vector store for storing and retrieving vectors
Hugging Face Transformers	Free	Library of pre-trained and fine-tuned transformer models for natural language processing (NLP), machine translation, and text generation
MS Office 365	Student	To develop or edit documents
Student	Student	Group Meetings
JIRA	Free	To build a project management plan
Canvas	Free	To develop presentations
Git Hub	Free	Standard repository to store and monitor the project progress
Figma	Free	To draw WBS and PERT Chart

Pinecone Vector. As mentioned above, vectors are stored and retrieved with the help of Pinecone Vector, which acts as the vector store for this project. Despite being a commercial tool, Pinecone Vector is a wise investment due to its potential. Explore their website and register for a free trial to obtain the Pinecone Vector license. Access to a small selection of vectors and features is available during this trial time. There is a choice to upgrade to the subscription plan if this project requires more comprehensive vector storage or further features.

Hugging Face Transformers. This is a library that gives this project access to transformer models that have already been trained and perfected, improving the natural language processing (NLP) abilities. Hugging Face Transformers, like Llama 2, is an open-source project, giving us the freedom to use it for a variety of objectives, including commercial use. It's simple to obtain the Hugging Face Transformers license; just download the library from the Hugging Face website. It's important to note that this library is accessible in a variety of programming languages, including Python, PyTorch, and TensorFlow, guaranteeing compatibility with the requirements of this varied project portfolio.

Project Cost Estimation and Justification

Many important resources and costs are needed for this project. Use of Llama 2 and Mistral 7B are the fundamental LLMs, they are free. And a limited free trial period with prospective membership fees is provided by Pinecone Vector. And coming to Hugging Face Transformers it is a free resource that is necessary for model construction. But Platforms for cloud computing (like AWS, Azure, or GCP) could have monthly costs of \$100 to \$500 or more. GPU purchases, which range in price from \$200 to \$600 depending on the RAM and usage, are essential for effective model training. The cost of data for training may be free or subject to licensing fees. These efforts are in line with the objective of providing the machine learning community with a sophisticated Reading Understanding System.

Building an LLM requires significant investment, yet its value cannot be disputed. Through the automation of processes like customer service, content creation, and research, LLMs have the potential to disrupt a number of industries. They also act as an innovation catalyst, facilitating the

creation of novel products and services. Even though the expenses can be high, this project will benefit from the effort due to the efficiency, productivity, and creativity gains.

Table 9

Cost Estimation and Justification

Tool or Resource	Type	Duration	Estimated Cost
Llama 2	Model (Software)	One-time	Free(Through Huggingface)
Mistral 7B	Model (Software)	One-time	Free(Through Huggingface)
GPT 3.5(For evaluation)	Model (Software)	Token wise	\$10-\$100
Pinecone Vector	Vector store (Software)	5 months	\$200-\$400
Cloud computing platform	Infrastructure (Hardware)	3 months	\$100-\$500
GPU for training the LLM	Infrastructure(Hardware)	5 months	\$200-\$600
			Total=\$500-\$1500

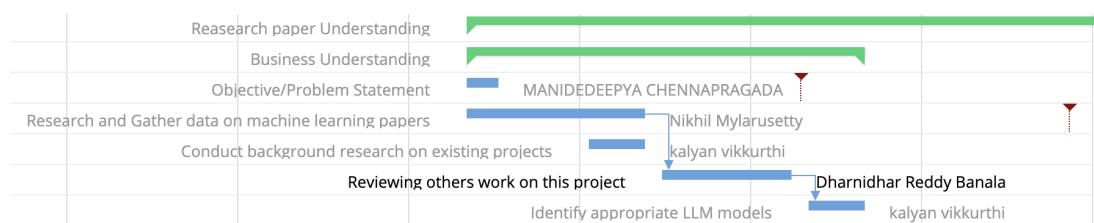
2.5 Project Schedule

Gantt Chart

Business Understanding. In this phase, the business level understanding happens where all the brainstorming for the problem statement happens and also how effectively solving the problem statement impacts the real world was assessed. Moreover, related work such as Background research, and existing technologies are also referred to have clear picture of the project execution.

Figure 4

Gantt Chart for Business Understanding

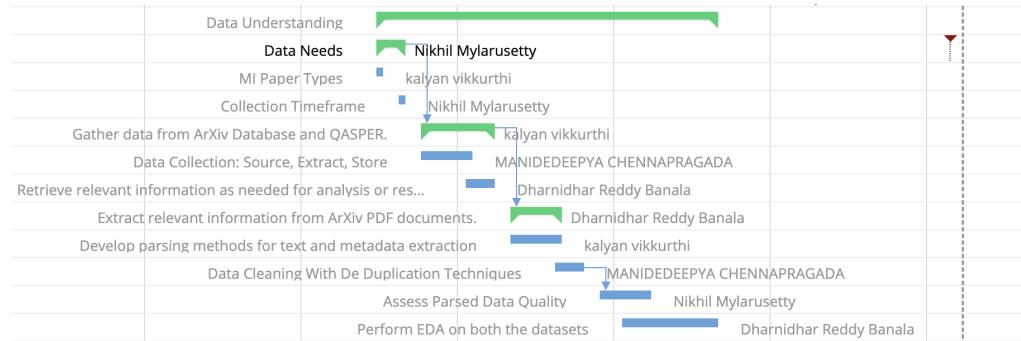


Note. Timelines, Tasks and Progress in Business Understanding are depicted in a Gantt chart.

Data Understanding. This is the second phase of the process where the raw data collection tasks are done. Apart from collection, the data cleaning is also done with deduplication techniques. Moreover, Exploratory data analysis is also performed in this step

Figure 5

Gantt Chart for Data Understanding

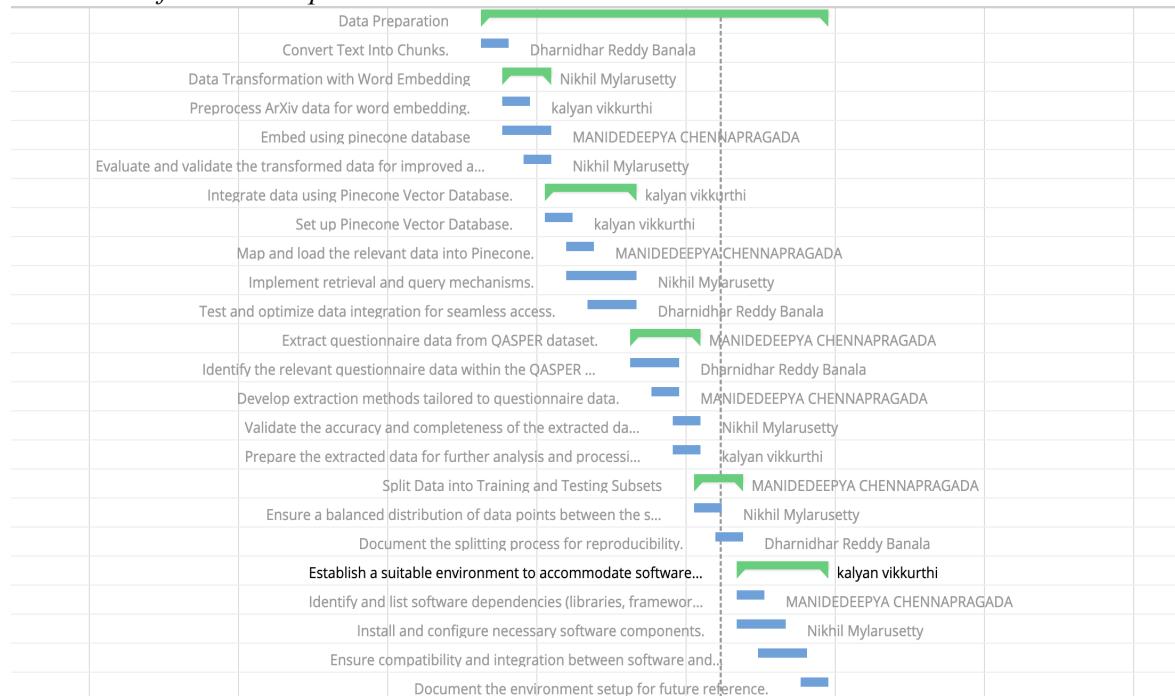


Note. Timelines, Tasks and Progress in Data Understanding are depicted in a Gantt chart.

Data Preparation. As the data for this project is in the form of pdf, the raw pdf must be processed in order to be trained on LLM Model. For this purpose, the data is divided into chunks and it is stored in Vector database for RAG Approach. And for the second approach the LLM Model is trained on QASPER Dataset

Figure 6

Gantt Chart for Data Preparation

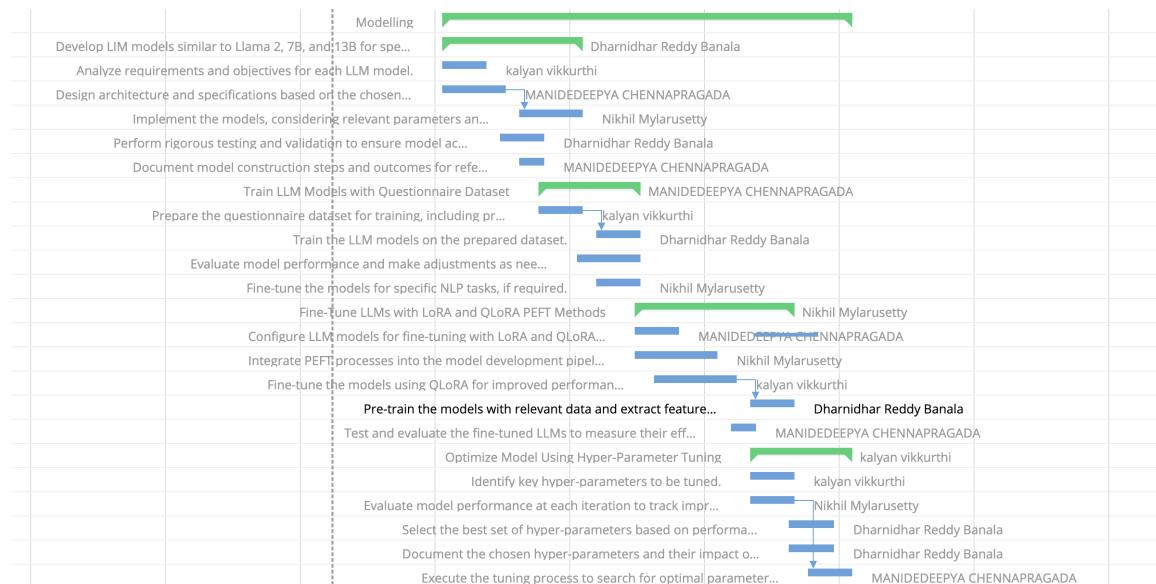


Note. Timelines, Tasks and progress in Data Preparation are depicted in a Gantt chart.

Modeling. In the modeling phase, the LLM Model is trained using PEFT Techniques where all the parameters are not tuned which is a cost-effective process with good results. Moreover, the models used are Llama 7B, 13B parameter models to train on research papers with PEFT(QLORA and LORA) techniques. For this model the data is taken from the previous cleaned and processed data.

Figure 7

Gantt Chart for Modeling

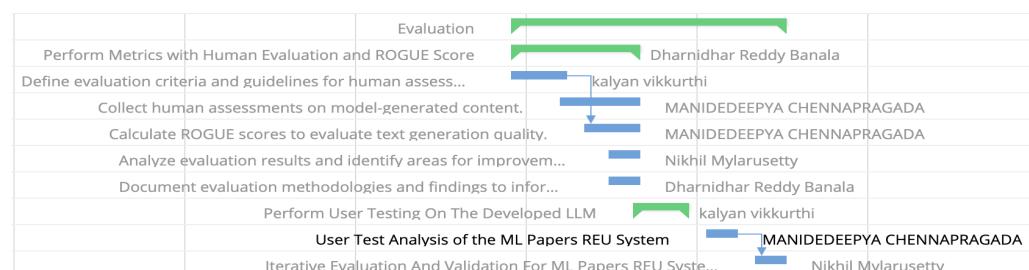


Note. Timelines, Tasks and Progress in Modeling are depicted in a Gantt chart.

Evaluation. The above modeling phase is evaluated with various techniques that include Collecting human assessments on model-generated content, calculating ROGUE Score for text generation quality. Moreover, in case of research papers the trained model is compared with real world CHATGPT, BARD, etc and measure how accurately the paper is read.

Figure 8

Gantt Chart for Evaluation

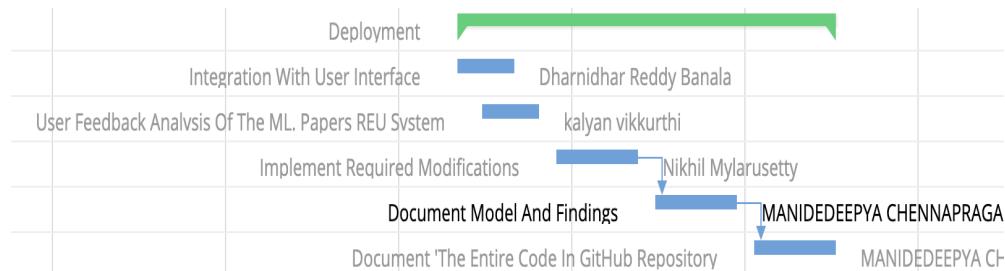


Note. Timelines, Tasks and Progress in Evaluation are depicted in a Gantt chart.

Deployment. Finally, after successful implementation of modeling phase. The User interface is built using streamlit in order to take the pdf from the user and give the accurate results in shorter manner.

Figure 9

Gantt Chart for Deployment



Note. Timelines, Tasks and Progress in Deployment Phase are depicted in a Gantt chart.

PERT Chart

The Figure 10 provides a PERT chart that outlines a comprehensive project timeline with various tasks and activities. This project has a timeline of approximately 6 months. The project begins with initial phases such as problem definition, data requirements identification, and data collection, which set the foundation for subsequent work. Key activities involve data conversion, fine-tuning a Large Language Model (LLM), data integration, system evaluation, and user testing.

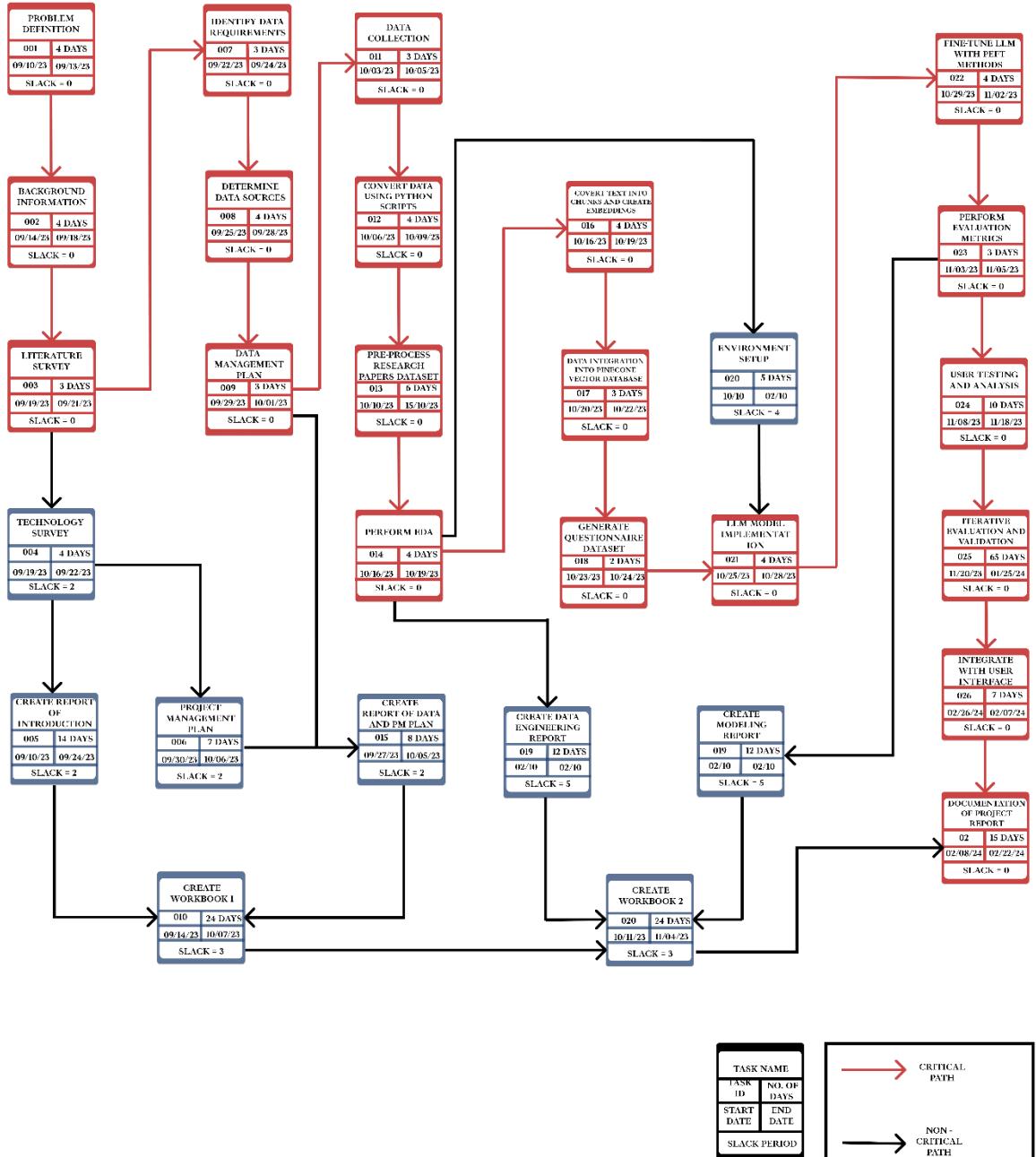
The project also includes various report creation tasks and documentation efforts, highlighting the importance of clear project communication. Notably, there is flexibility in certain activities, as denoted by slack time, allowing some leeway without affecting the project's overall schedule.

The project concludes with a significant phase of integrating with the user interface, workbook creation, and final documentation. Overall, the PERT chart provides a structured overview of the project's timeline and interdependencies.

Figure 10

PERT Chart

PERT CHART



Data Engineering

3.1 Data Process

Data Process for PDFs of ArXiv

In order to further their research, the group needed to obtain and prepare PDF files from the renowned arXiv website, a repository of academic research articles. To help with this, a script has been written that will sort through a CSV list of arXiv IDs. This application will be identifying each ID and download the corresponding PDFs from arXiv. Once these are obtained, the PDFs were meticulously organized and archived in a designated Google Drive folder, ensuring a centralized repository for the essential research papers.

The process is followed by the next stage, where the focus will be on integrating the stored PDFs with the Colab environment, which will be an essential for the research workflow. These PDFs are moved from the local machine into the Google drive for central storage. The Colab workspace will be connected with the Google Drive directory. This connectivity will be a critical step in allowing for seamless data manipulation and processing.

However, raw PDFs, in their unstructured form, posed many challenges. So to overcome this the team members decided on transforming these indistinct texts into an organized format ready for research. This transformation involves extracting two critical components from the PDFs: metadata and content. Metadata, the skeletal framework of each PDF, contained fundamental details such as titles, authorship, and publication dates, and served as a guidepost for document arrangement.

On the other hand, the textual reservoir, the heart and soul of the papers, was brimming with ideas and information. To accomplish this extraction, the team used the PyPDF2 library, an advanced package that disassembles PDFs and extracts both metadata and content. Following extraction, the metadata and content were methodically formatted, frequently in JSON format.

QASPER Data Processing

The researchers created a dataset that included information such as the paper's ID, title, abstract, and full-text parts with their appropriate titles, among other things. Details on the person who formulated the question were documented within the "qas" segment. This covers their comprehension of the paper's topic, whether they read the paper, and the search terms they used to find the abstract. There were distinctions established between solvable and insolvable issues for answers. Solvable answers could be straight quotes from the paper, free-form responses, or simple yes/no answers supported by evidence. There was also an area for the answer providers to underline evidence.

The QASPER data has been checked for null values and duplicates once it was acquired. Also modified the data by extracting the content from a column labeled 'qas' which has data in the form of json by writing a script. The data will be split after all of the exploration and transformations are made for training and another for testing. This separation was critical in preparing the data for the next steps in our research.

Figure 11

Data Process for PDF Dataset

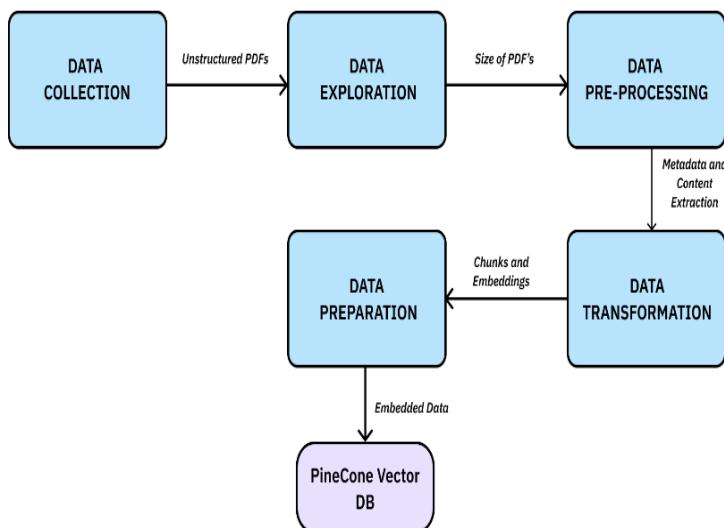
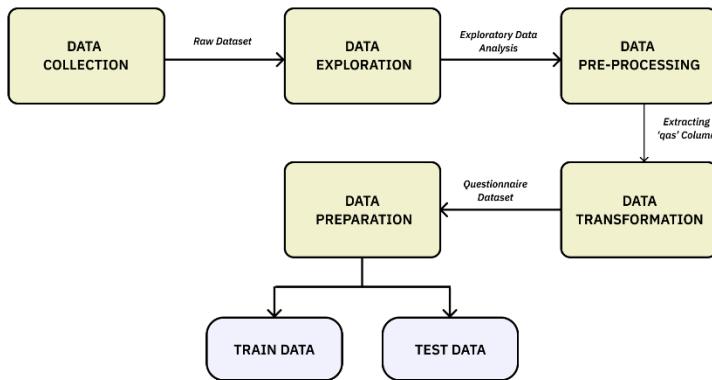


Figure 12*Data Process for QASPER Dataset*

3.2 Data Collection

The data collection for this project involved taking data from two different sources having two main approaches. The data for the first approach was taken from the arXiv database using the id of the PDF. These PDFs are approximately 1000 having a specific domain as NLP which are downloaded using an automated process.

The data for the second approach which is fine-tuning the model using a questionnaire dataset is taken from Allen Institute for AI which is the QASPER dataset. The collection process of QASPER data by the original collector details the approach to data gathering was built around using two primary tools. The initial tool is the Question Generation Tool, which enables the users to craft questions based on the paper titles and abstracts for each paper. Complementing this, the Evidence and Response Finder offers users a pathway to pinpoint relevant evidence and draw answers to the earlier posed questions.

The paper selection is done by the team using the expansive S2ORC library which is known for the collection of digital open-access research manuscripts. To refine their paper selections, the researchers employed stringent criteria. They prioritized papers linked with arXiv database that had a corresponding LaTeX format. Their spotlight was firmly on computational linguistics, ensuring the content was both pertinent and of high standard. After

the selection, the next step involved was to transform the selected papers into a more user-friendly version which will be full-text. This was done by preserving essential structural components and even involving visual elements such as images and tables from the original papers in this dataset.

Table 10

Data Collection Plan for ArXiv and QASPER Datasets

Description of the data collection										
Where are we collecting the data from?	PDFs from arXiv: This dataset consists of unstructured PDF documents sourced from arXiv, a preprint repository renowned for hosting research papers in various scientific fields, including physics, mathematics, computer science, and more. These papers are typically cutting-edge studies uploaded by researchers prior to undergoing the official peer-review process, thereby offering a wealth of early insights into emerging scientific concepts and findings. QASPER from AllenAI: QASPER is a unique dataset curated by AllenAI that contains natural questions about scientific papers, along with corresponding answers derived directly from those papers. Designed to foster advancements in question-answering systems, especially in the context of scientific literature, this dataset presents a rich blend of user-generated questions and data-extracted answers, reflecting genuine research queries and comprehension tasks									
What will be done with the data once it has been collected?	The collected data, comprising research papers from the arXiv database and the QASPER dataset, will be utilized to develop a comprehensive system designed to enhance the efficiency and effectiveness of analyzing machine learning research papers. These papers will be integrated into the Pinecone vector database to optimize data storage and retrieval, ensuring swift searches and analyses. To bolster natural language processing and improve user interactions, a Large Language Model (LLM) will be formulated using tools like llama 2 and retrieval augmented generation. This model will undergo extensive training and fine-tuning to provide precise text comprehension and generation. The LLM's performance will be rigorously assessed through various metrics, such as ROUGE scores, to guarantee contextually accurate responses. Ultimately, this system aims to offer users an interactive platform where they can delve deeper into the content, enabling them to navigate and understand the complexities of ML papers through an intuitive questionnaire format.									
Key Variables - A summary of the chosen input variables (Y's) and/or output variables (X's)										
Variable title	1	2	3	4	5	6	7	8	9	10
Input (X) or output (Y) variable?	Metadata	Content								
Unit of measurement	X	X								
Data type	N/A	N/A								
Collection method	String									
If manual	Automated extraction from PDF									
Historical data exist?	N/A									
Source of historical data	Yes, based on publication date									
Sampling frequency	arXiv Database									
Sub-grouping needed?	N/A									
Sub-group size	No									
Data collector	Dharmidhar Reddy Banala									
Start date	N/A									
Due date	N/A									
Key Variables - A summary of the chosen input variables (Y's) and/or output variables (X's)										
Variable title	11	12	13	14	15	16	17	18	19	
Input (X) or output (Y) variable?	id	title	abstarct	Full_text	qas	Figures_and_table				
Unit of measurement	X	X	X	X	X	X				
Data type	N/A	N/A	N/A	N/A	N/A	N/A				
Collection method	String									
If manual	Automated Extraction, Human annotators									
Historical data exist?	N/A									
Source of historical data	Yes, for user-sourced questions									
Sampling frequency	N/A									
Sub-grouping needed?	Yes									
Sub-group size	Once									
Data collector	Various scientific papers and journals.									
Start date	N/A									
Due date	N/A									

The paper also uses a unique methodology of gathering data by the approach of creating questions first and then sourcing answers having a touch of authenticity to the

generated queries where the people behind this effort were no novices. The graduate students who are deeply into the NLP studies and experienced NLP professionals who were well-acquainted with the concepts of NLP literature. This hourly framework was adopted to emphasize the quality of content over its sheer quantity. While 25 of these experts framed the questions, a larger group of 51 dove into providing insightful answers. The question generation process was further fine-tuned with an interactive search tool, granting experts the autonomy to select papers they found intriguing, catering to both broad and specific academic curiosities.

Dataset Sample for the Raw Data Resource

QASPER Dataset

- id:
 - *Description:* This column represents the unique identifier assigned to each research paper.
 - *Details:* The format appears to be a combination of numbers, possibly indicating the year and a unique number assigned to the paper in that year (e.g., 1909.00694).
- title:
 - *Description:* This column represents the title of the respective research paper.
 - *Details:* It provides a brief description of the content or main focus of the paper (e.g., "Minimally Supervised Learning of Affective Events").
- abstract:
 - *Description:* This column contains the abstract of the research paper.
 - *Details:* An abstract is a concise summary that provides a snapshot of the main points and findings of the paper. It is the first section that readers see, offering a brief overview of the research.

- full_text:
 - *Description:* This column provides sections of the full text of the research paper.
 - *Details:* This dataset divides the paper into different sections such as 'Introduction', 'Related Work', and so on. It seems to provide a dictionary-like structure with section names and their respective content.
- qas:
 - *Description:* This column is focused on the questions and answers associated with the research paper.
 - *Details:* It appears to have a dictionary-like structure. Questions related to the paper are stored with their respective answers. The example showcases questions like 'What is the seed lexicon?' for a paper, though the answer isn't immediately visible in the sample.
- figures_and_tables:
 - *Description:* This column provides information related to figures and tables included in the research paper.
 - *Details:* It appears to use a dictionary-like structure, capturing captions or descriptions of the figures or tables. For instance, there's a caption 'Figure 1: An overview of our method...' associated with the first paper

PDF Downloading from arXiv and Storing in Google Drive

There is a need for several PDF files from arXiv, and each PDF was associated with a different arXiv ID. We developed a specific software that read a list of arXiv IDs (in a CSV file) to obtain this data. It browsed the list, located each ID, and requested the corresponding PDF from arXiv. Then, we arranged and stored all of these downloaded PDFs in a particular

Google Drive folder. This was the first and most critical step since it ensured that we have all the necessary key PDFs in one location.

Now that the PDFs are safely stored in Google Drive, the next important step was to make these files available in the Colab environment. We did this by connecting Google Drive to Colab and setting up the right file paths. This connection was crucial because it allowed us to easily work with the PDFs in our research project. When we connected Google Drive to Colab, it basically made our stored files accessible in the Colab environment, where we did our data processing.

Figure 13

Downloaded PDFs in Local Machine

```
→ Saved 1909.00694.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.07723.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1705.09665.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1908.06606.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1811.00942.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1805.02400.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1907.05664.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1910.14497.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1912.02481.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1810.04528.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2002.02224.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.07433.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.12218.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1904.09178.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.06651.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1910.04269.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1906.00378.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1912.13072.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1712.09127.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2001.00137.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1910.03042.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2002.06644.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1809.08731.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1707.00995.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1809.04960.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1909.08402.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1909.11189.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1810.05320.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.08529.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1708.05873.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 2003.08553.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
Saved 1909.09491.pdf to /content/drive/MyDrive/Arxiv_Qasper_dataset
```

Figure 14

Downloaded PDFs Stored into Google Drive

 1601.00901.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 879 KB	    
 1601.01705.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 3 MB	
 1601.02166.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 88 KB	
 1601.02543.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 286 KB	
 1601.04012.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 757 KB	
 1601.06068.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 355 KB	
 1601.06081.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 225 KB	
 1601.06738.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 588 KB	
 1602.00812.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 346 KB	
 1602.01208.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 1.8 MB	
 1602.01595.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 358 KB	
 1602.03661.pdf 	 Dharnidhar reddy banala	Oct 20, 2023	Dharnidhar re... 385 KB	

QASPER Dataset

The compiled dataset encompassing varied facets like the paper's ID, title, abstract, and full-text sections along with their designated headings, among other aspects. Significantly, within the "qas" segment, we documented details concerning the background of the person who formulated the question, their grasp on the paper's topic, whether they had reviewed the paper, and the search terms used to retrieve the paper's abstract. Regarding answers, we could pinpoint if a question was unresolvable or not. If solvable, the answer could be an extract from the paper, a freely composed response, or a simple yes/no, supported with evidence. There was also a section where the answer givers could highlight the evidence they provided. This data collection was earmarked for both training and validation purposes, containing 888 papers with 2593 questions for training, and 281 papers with 1005 questions for validation, each accompanied by a respective set of answers.

In this venture, our goal was to amass the QASPER dataset from Hugging Face. We kicked off by bringing in the necessary libraries, then formed a bridge between our Google Drive and the Colab platform using the `google.colab.drive.mount()` function. This enabled us to save the dataset straight to our Google Drive, bypassing the need to download it onto our personal systems. We got hold of the crucial dataset using the `datasets.load_dataset` function. Following this, we made sure the dataset was both secure and easily reachable for our specially fine-tuned Llama and Mistral 7B models by saving it directly to our Google Drive. This tactic smoothed out the process of collecting the dataset and ensured its availability for the goals of our project.

Figure 15

Columns in the QASPER Dataset

```
DatasetDict({
    train: Dataset({
        features: ['id', 'title', 'abstract', 'full_text', 'qas', 'figures_and_tables'],
    }),
    test: Dataset({
        features: ['id', 'title', 'abstract', 'full_text', 'qas', 'figures_and_tables'],
    })
})
```

Figure 16

Sample QASPER Raw Dataset

id	title	abstract	full_text	qas	figures_and_tables
0	1909.00694	Minimally Supervised Learning of Affective Events	Recognizing affective events that trigger positive ...	{'question': ['What is the seed lexicon?', 'What is the distribution of the lexicon?']}	{'caption': ['Figure 1: An overview of our method. The figure shows a flowchart of the system architecture. It starts with "Input Data" leading to "Feature Extraction" and "Sentiment Analysis". These feed into "Event Detection" and "Entity Recognition". Finally, "Event Detection" leads to "Output: Event-Entity pairs".']}
1	2003.07723	PO-EMO: Conceptualization, Annotation, and Model...	Most approaches to emotion analysis regarding ...	{'question': ['Does the paper report macro F1...']}	{'caption': ['Figure 1: Temporal distribution ...']}
2	1705.09665	Community Identity and User Engagement in a Multi...	A community's identity defines and shapes its ...	{'question': ['Do they report results only on ...']}	{'caption': ['Figure 1: A: Within a community ...']}
3	1908.06606	Question Answering based Clinical Text Structur...	Clinical text structuring is a critical and fu...	{'question': ['What data is the language model...']}	{'caption': ['Fig. 1. An illustrative example ...']}
4	1811.00942	Progress and Tradeoffs in Neural Language Models	In recent years, we have witnessed a dramatic ...	{'question': ['What aspects have been compared...']}	{'caption': ['Table 1: Comparison of neural la...']}
...
883	1702.03274	Hybrid Code Networks: practical and efficient ...	End-to-end learning of recurrent neural netwro...	{'question': ['Does the latent dialogue state ...']}	{'caption': ['Figure 1: Operational loop. Trap...']}
884	1610.03112	Leveraging Recurrent Neural Networks for Multi...	Social norms are shared rules that govern and ...	{'question': ['Does this paper propose a new t...']}	{'caption': ['Table 1: Statistics of the corpora...']}
885	1607.03542	Open-Vocabulary Semantic Parsing with both Dis...	Traditional semantic parsers map language onto...	{'question': ['What knowledge base do they use...']}	{'caption': ['Figure 1: Overview of the component...']}
886	1812.10860	Can You Tell Me How to Get Past Sesame Street?...	Natural language understanding has recently se...	{'question': ['Do some pretraining objectives ...']}	{'caption': ['Figure 1: Our common model design...']}
887	1712.02121	A Novel Embedding Model for Knowledge Base Com...	In this paper, we propose a novel embedding mo...	{'question': ['Did the authors try stacking mu...']}	{'caption': ['Table 1: The score functions in ...']}

888 rows × 6 columns

Figure 17

Null Value Count in QASPER Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 888 entries, 0 to 887
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               888 non-null    object 
 1   title            888 non-null    object 
 2   abstract         888 non-null    object 
 3   full_text        888 non-null    object 
 4   qas              888 non-null    object 
 5   figures_and_tables 888 non-null  object 
dtypes: object(6)
memory usage: 41.8+ KB
```

Figure 18

Presence of Duplicates in QASPER Dataset

```
'full_text' contains dictionaries.  
'qas' contains dictionaries.  
'figures_and_tables' contains dictionaries.
```

```
0      False
1      False
2      False
3      False
4      False
       ...
883     False
884     False
885     False
886     False
887     False
Length: 888, dtype: bool
```

Figure 19

Word Cloud for Abstract Column

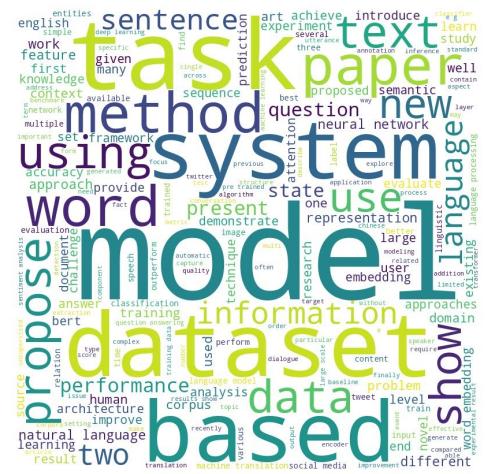


Figure 20

Word cloud for Title Column

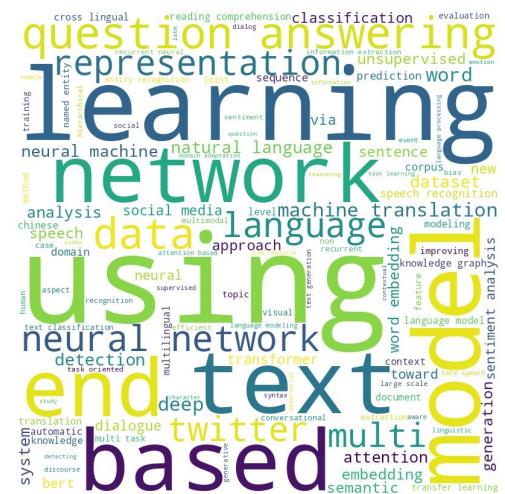


Figure 21

Lexical Dispersion Plotting for Title Column

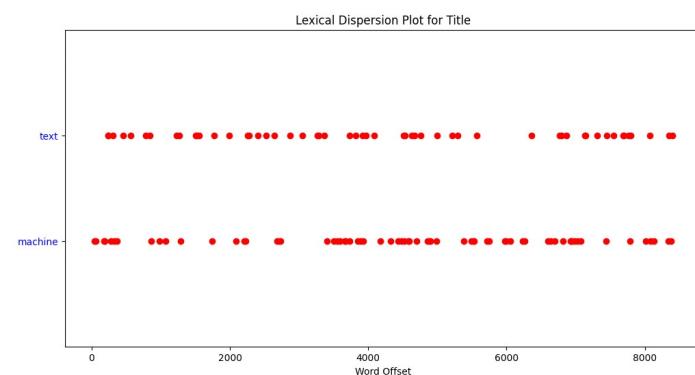
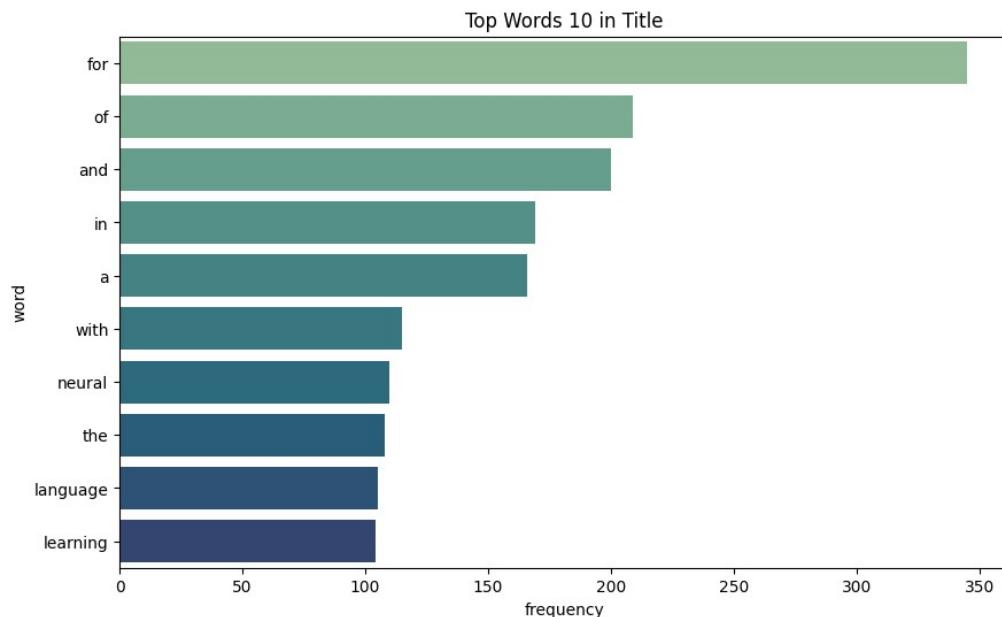


Figure 22

Top-10 words in Title of PDFs



3.3 Data Pre-processing

Extracting the Unstructured PDF to Metadata and Content

The main goal of our PDF data preparation was to change these messy documents into organized data that we could use for research and analysis. We did this by extracting two key things from the PDFs

Metadata. This is like the basic info about each PDF, such as its title, the author(s), when it was published, and other details that help us sort and arrange the PDFs. Getting this metadata was really important because it helped us sort and find our documents easily.

Content. This is the text that appears in the PDF files which has the real meaning of the papers. Obtaining this literature was maybe the most important phase because it provided us with the necessary knowledge and information included in the PDFs.. With this text, we could do different kinds of analysis, like studying the words and sentences, or using tools for language and text understanding.

To do this, we used a library called PyPDF2, which helped us take apart the PDFs and get out the metadata and content. After we got this information, we carefully saved it.

Metadata might be saved in a structured way, like in a JSON file, while content might be kept as plain text files.

Figure 23

Snippet of the Metadata and Content as a DataFrame

	/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.01705.pdf	/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.06068.pdf	/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.02166.pdf
metadata	{'Author': '', 'CreationDate': 'D:2016060900...'}	{'Author': '', 'CreationDate': 'D:2016080800...'}	{'Producer': 'dvips + GPL Ghostscript GIT PRE...'}
content	Learning to Compose Neural Networks for Questi...	Paraphrase Generation from Latent-Variable PCF...	arXiv:1601.02166v1 [cs.CL] 9 Jan 2016Empiric...
2 rows × 886 columns			

Figure 24

Snippet of Transposed dataset of the Metadata and Content

	metadata	content
/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.01705.pdf	{'Author': '', 'CreationDate': 'D:2016060900...'}	Learning to Compose Neural Networks for Questi...
/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.06068.pdf	{'Author': '', 'CreationDate': 'D:2016080800...'}	Paraphrase Generation from Latent-Variable PCF...
/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.02166.pdf	{'Producer': 'dvips + GPL Ghostscript GIT PRE...'}	arXiv:1601.02166v1 [cs.CL] 9 Jan 2016Empiric...
/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.06738.pdf	{'Author': '', 'CreationDate': 'D:2022030100...'}	A Label Semantics Approach to Linguistic Hedge...
/content/drive/MyDrive/Set1_100_Papers/Copy of 1601.04012.pdf	{'CreationDate': 'D:20160118013640Z', 'Creat...'}	Detecting and Extracting Events from TextInDoc...

Missing values

This project requires a replacement plan when there are chances of gaps in the pdf content or the meta data, so can enable from predicting better accurate predictions for such missing evidence based on the information that is already accessible. So this kind of problems might affect this replacement approaches when there are huge gaps in both the data and meta data.

Handling Null Values

When loading PDFs, we ran into problems, particularly when the PDFs don't have the right layout or documentation. Recognizing these issues and choosing a course of action are necessary when dealing with null values. It's essential to have a strategy for dealing with

unreadable or troublesome PDFs. This strategy can entail skipping troublesome files, recording faults for later use, or, if possible, trying to fix the PDF structure.

Figure 25

Unreadable PDF Files

 1611.01400.pdf	 me	Oct 20, 2023 me	707 KB	
 1611.02550.pdf	 me	Oct 20, 2023 me	570 KB	
 1611.03382.pdf	 me	Oct 20, 2023 me	1.2 MB	
 1611.04361.pdf	 me	Oct 20, 2023 me	304 KB	
 1611.04642.pdf	 me	Oct 20, 2023 me	1.7 MB	
 1611.09441.pdf	 me	Oct 20, 2023 me	108 KB	
 1612.04675.pdf	 me	Oct 20, 2023 me	8 KB	
 1612.05270.pdf	 me	Oct 20, 2023 me	394 KB	
 1612.05310.pdf	 me	Oct 20, 2023 me	101 KB	
 1612.06897.pdf	 me	Oct 20, 2023 me	116 KB	
 1612.07843.pdf	 me	Oct 20, 2023 me	2.4 MB	

So, in the above picture, there is a blue color corrupted file, which do not have the content and it seems to be a null valued document which is been removed using a function named PyPDFError which eliminates such files.

Elimination of Duplicates

There were few duplicate papers in the directory which are been removed as they might skew and duplicate the content. To guarantee that each paper is unique and delivers unique information to the reading understanding system, the preprocessing phase for duplicates entails the identification and deletion of such repetitive content. So for that we have used MD5 hash values to identify such duplicate pdfs.

Extracting ‘qas’ Column from QASPER

The ‘qas’ column in the dataset represents the questions (and potentially answers) associated with each research paper. In the dataset this column appears to have a dictionary i.e a JSON-like structure, with keys such as question that indicate the content. So the column has to extracted for further transformation of the data.

Figure 26

Extracted 'qas' Column from Raw Dataset

```
qas
0  {'question': ['What is the seed lexicon?', 'Wh...
1  {'question': ['Does the paper report macro F1?...
2  {'question': ['Do they report results only on ...
3  {'question': ['What data is the language model...
4  {'question': ['What aspects have been compared...
...
883 {'question': ['Does the latent dialogue state ...
884 {'question': ['Does this paper propose a new t...
885 {'question': ['What knowledge base do they use...
886 {'question': ['Do some pretraining objectives ...
887 {'question': ['Did the authors try stacking mu...
[888 rows x 1 columns]
```

3.4 Data Transformation:

In this part, we'll explain the important data transformation steps used in the project.

These steps are vital because they help our Reading Understanding System work effectively with Machine Learning-related technical papers and articles.

We first divide PDF files into smaller, simpler text chunks. This facilitates our system's handling and understanding of the text. And next the system may process and retrieve information more effectively by using embeddings, which are essentially representations of the text. The system relies heavily on these embeddings to comprehend the technical papers and articles related to machine learning.

Dividing PDFs into Chunks

An enormous problem is posed by the huge amount of information in lengthy technical articles. To solve this problem, we developed a method for breaking up these PDF files into smaller, simpler-to-understand "chunks." This makes it easier to break down and to interact with the content.

Our approach involves using a tool called 'PyPDFDirectoryLoader' from the 'langchain' framework. This tool enables us to access and load more than 1,000 PDF documents stored in a specified directory. These PDFs, with a total chunk size of 200 for all of them, are now ready for text processing.

Used a function called the "RecursiveCharacterTextSplitter" to divide these massive documents into smaller, more manageable chunks. This text splitting procedure is governed by a set of rules: the chunk size is 200 characters, with a 20-character overlap between chunks. This division provides a staggering 195,000 text chunks. These chunks are structured to provide a manageable size for further processing without leaving out any crucial information.

The outcome of this process is a collection of text chunks, each containing a portion of the original PDF content. These chunks are now ready for additional transformation and analysis.

Figure 27

Example Chunk Information

```
Document(page_content='as a dataset. These sentences are already annotated with links to other pages, which\nare also instances of DBpe
{'source': '/content/drive/MyDrive/10_papers_arxiv_gasper_test/Copy of 1601.00901.pdf', 'page': 1})
```

Creating Embeddings

A crucial next step is the conversion of the text chunks into numerical vectors called embeddings. These embeddings are essential in the processing of textual information because they maintain the context and underlying meaning of the text.

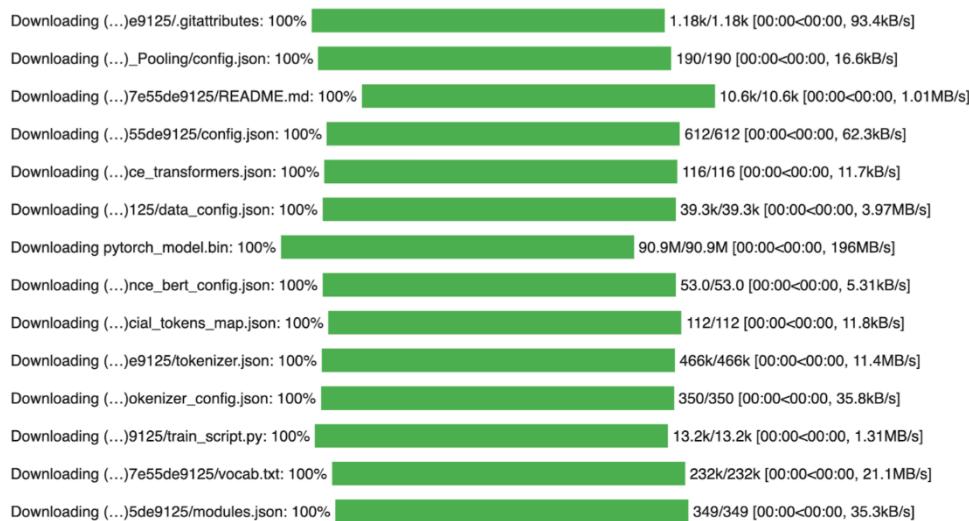
Because they help computers comprehend and interact with language in a more meaningful way, embeddings serve as a kind of numerical representation of the text's core. Built these embeddings using the 'HuggingFaceEmbeddings' package, which uses the incredibly complex pre-trained language model'sentence-transformers/all-MiniLM-L6-v2.'

These embeddings are created, managed, and saved in a software program called Pinecone's vector store. Pinecone is crucial in helping us locate and retrieve relevant data fast when a user gives a query. This also enables to match user queries with the most relevant text chunks because of the fact that all of the text chunks are maintained in this ordered vector approach.

As a result, embeddings are essential because they convert language into computer-understandable form, which is necessary for our system to process and retrieve information properly. Making text understandable to computers is necessary to ensure that everything works together. The below figure shows the pre trained models libraries.

Figure 28

Transformation of Chunks to Embeddings



3.4.3 Qasper Dataset

In pursuit of this, we crafted a Python script that effectively transitioned the unprocessed data into a structured DataFrame. This step significantly improved our data processing and analytical prowess. This transformation allowed us to methodically extract vital attributes such as paper IDs, pertinent questions, their nature (binary or descriptive), respective answers, and the supporting evidence, including highlighted segments.

Our data was initially bifurcated into training and validation subsets, encapsulated within JSON files. To process this, we meticulously looped through each file. Every paper became our focal point as we sifted through its associated queries and responses. By sidelining ambiguous questions, we zoned in on those with definitive answers.

This enabled us to cull essential details like the paper IDs, questions, their categorization (either binary or free-form), the corresponding answers, the evidence

substantiating those answers, and highlighted references. The essence of this transformation lay in its ability to declutter and systematize the dataset. By doing so, we were empowered to access targeted information like answers paired with their supporting evidence in a structured fashion. Such an organized framework immensely expedited our research trajectory, equipping us to derive meaningful interpretations from the dataset. In essence, this rigorous refinement phase was pivotal, setting the stage for harnessing the dataset to its utmost potential for our research aspirations.

Figure 29

Transformed QASPER Dataset

Paper ID	Question	Yes/No Question	Answer	Evidence	Highlighted Evidence
0 1909.00694	What is the seed lexicon?	None	a vocabulary of positive and negative predicat...	The seed lexicon consists of positive and nega...	The seed lexicon consists of positive and nega...
1 1909.00694	What are the results?	None	Using all data to train: AL – BiGRU achieved ...	FLOAT SELECTED: Table 3: Performance of variou...	FLOAT SELECTED: Table 3: Performance of variou...
2 1909.00694	How are relations used to propagate polarity?	None	based on the relation between events, the sugg...	In this paper, we propose a simple and effect...	As illustrated in Figure FIGREF1, our key idea...
3 1909.00694	How big is the Japanese data?	None	700000 pairs of events were extracted from th...	As a raw corpus, we used a Japanese web corpus...	As a raw corpus, we used a Japanese web corpus...
4 1909.00694	What are labels available in dataset for super...	None	No	Affective events BiBREFO are events that typi...	In this paper, we work on recognizing the pola...
...
3046 1701.05574	What is the best reported system?	None	Gaze Sarcasm using Multi Instance Logistic Reg...	FLOAT SELECTED: Table 3: Classification result...	FLOAT SELECTED: Table 3: Classification result...
3047 1701.05574	What cognitive features are used?	None	Readability (RED), Number of Words (LEN), Avg...	FLOAT SELECTED: Table 2: The complete set of f...	FLOAT SELECTED: Table 2: The complete set of f...
3048 1907.01468	What approaches do they use towards text analy...	None	No	This contrasts with much of the work in comput...	The approaches we use and what we mean by 'suc...
3049 1907.01468	Do they demonstrate why interdisciplinary insi...	False	No		
3050 1907.01468	What kind of issues (that are not on the foref...	None	No	We typically start by identifying the question...	We typically start by identifying the question...

3051 rows x 6 columns

Incorporation of Contextual References. The transformation process began with the augmentation of context to the questions. Originally, the dataset's questions were standalone, lacking references to the source papers they were derived from. To remedy this, the phrase "In this paper, [Paper Name]" was programmatically prepended to each question, providing a clear linkage to the originating scholarly work. This contextualization is critical as it allows the LLM to understand the provenance of the information, potentially impacting the model's accuracy in answering or referencing the questions.

Synthesis of Meaningful Answers. The second image delineates the subsequent phase where answers were synthesized from multiple columns to form a singular, comprehensive response. This intricate process involved selecting the most cogent pieces of information from the Answer, Yes/No, Evidence, and Highlighted Evidence columns. By leveraging these fragments, the transformation script crafted a coherent and meaningful answer, thereby enhancing the quality of the dataset. The resulting answers were not only complete but also tailored to encapsulate the essence of the evidence presented.

Figure 30

Transformed Data after changes.

Merged Title and Question	Answer
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What is the seed lexicon?	The seed lexicon consists of positive and negative predicates. If the predicate of an extracted event is in the seed lexicon, then it is considered a match.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What are the results?	The results show that the proposed model outperforms baseline models on several benchmarks.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How are relations used to propagate polarity?	Relations are used to propagate polarity by linking adjacent words in a sentence. If a word is positive or negative, its polarity is propagated to the next word in the sequence.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big is the Japanese data?	The Japanese dataset contains approximately 10,000 annotated examples.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What are labels available in dataset for supervision?	Labels available in the dataset include positive, negative, and neutral sentiment categories.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big are improvements of supervised learning results trained on small datasets?	Improvements of supervised learning results trained on small datasets are significant, often exceeding 50% accuracy.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How does their model learn using mostly raw data?	The model learns using mostly raw data by extracting features from the text and using them to predict sentiment.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What are the aesthetic emotions formalized?	The aesthetic emotions formalized include beauty, ugliness, and symmetry.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big is seed lexicon used for training?	The seed lexicon used for training contains approximately 100 words.
In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How large is raw corpus used for training?	The raw corpus used for training contains millions of sentences.
In the paper PO-EMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry, Does the paper report macro F1?	The paper reports a macro F1 score of approximately 85%.
In the paper PO-EMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry, How is the annotation experiment evaluated?	The annotation experiment is evaluated using a confusion matrix.
In the paper PO-EMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry, What are the aesthetic emotions formalized?	The aesthetic emotions formalized include beauty, ugliness, and symmetry.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, Do they report results only on English data?	No, they report results on multiple languages.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, How do they report results on English data?	They report results on English data separately from other languages.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, How do the various social phenomena examined manifest in different types of communities?	Various social phenomena such as user engagement, community size, and communication patterns manifest differently across different types of communities.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, What patterns do they observe about how user engagement varies with time?	They find that user engagement is highly dynamic and constantly shifts between different time windows.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, How did the select the 300 Reddit communities for comparison?	They selected the top 300 most active and diverse communities on Reddit.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, How do the authors measure how temporally dynamic a community is?	They measure temporal dynamics by tracking user activity over time.
In the paper Community Identity and User Engagement in a Multi-Community Landscape, How do the authors measure how distinctive a community is?	They measure distinctiveness by comparing user activity across different communities.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, What data is the language model pretrained on?	The language model is pretrained on a large dataset of clinical text.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, What is the proposed model compared against?	The proposed model is compared against baseline models.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, How is the clinical text structuring task defined?	The clinical text structuring task involves identifying specific clinical entities and their relationships in text.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, What are the specific tasks being unified?	The specific tasks being unified include entity extraction, relation extraction, and classification.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, Is all text in this dataset a question, or are there unrelated sentences?	Not all text in the dataset is a question; some are unrelated sentences.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, Our dataset is annotated based on Chinese pathology reports provided by the Department of Gastrointestinal Surgery.	The dataset is annotated based on Chinese pathology reports provided by the Department of Gastrointestinal Surgery.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, How many questions are there in the dataset?	The dataset contains thousands of questions.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, Our dataset is annotated based on Chinese pathology reports provided by the Department of Gastrointestinal Surgery.	The dataset is annotated based on Chinese pathology reports provided by the Department of Gastrointestinal Surgery.
In the paper Question Answering based Clinical Text Structuring Using Pre-trained Language Model, How they introduce domain-specific features into pre-trained language models?	They introduce domain-specific features into pre-trained language models by fine-tuning them on clinical text datasets.

3.5 Data Preparation

VectorDB Storage of Embeddings

The data preparation phase is crucial in ensuring that the Reading Understanding System has easy access to a sizable library of scientific articles in PDF format for processing and retrieval. Important parts of the data preparation includes the division of PDF files into meaningful text chunks, the creation of embeddings, and their subsequent storage in a Vector Database, more often referred to as VectorDB, for rapid and intelligent access.

These text chunks are produced as the first phase, using a method intended to find a balance between granularity and comprehensiveness as previously mentioned. Each chunk is carefully created to have 200 characters or less, ensuring that it has enough context to be

meaningful. A 20-character overlap between portions is used to preserve cohesiveness. These pieces act as the foundation for additional processing.

The construction of embeddings is the next critical step after having these text chunks.

The core of the content contained within the chunks is captured by these number representations. To do this, we use the 'sentence-transformers/all-MiniLM-L6-v2' from Hugging Face, an efficient pre-trained language model. With this method, text may be vectorized to create representations of the text's content that contain both semantic and contextual information.

Figure 31

Output for Stored Embeddings

```
[Document(page_content='parsing, ontology learning, grammar induction and others.\n5.1. Semantic parsing\nThe goal of semantic parsing is to map text to meaning repres
Document(page_content='parsing, ontology learning, grammar induction and others.\n5.1. Semantic parsing\nThe goal of semantic parsing is to map text to meaning repres
Document(page_content='parsing, ontology learning, grammar induction and others.\n5.1. Semantic parsing\nThe goal of semantic parsing is to map text to meaning repres
Document(page_content='parsing, ontology learning, grammar induction and others.\n5.1. Semantic parsing\nThe goal of semantic parsing is to map text to meaning repres
```

The use of Vector Database(VectorDB), is a significant stage in the data preparation process. The text chunks' numerical representations, or embeddings, are what it is mostly used for. To ensure that our data is securely stored and easy to access for coming retrieval procedures, this step is essential. For managing this process, we used Pinecone, a solid software designed specifically for handling numerical data efficiently. Pinecone makes storing and retrieving these vector representations straightforward.

Another crucial step in this phase is a thorough check that our text chunks have been successfully placed in the VectorDB. This verification procedure has allowed the vector representations of our text chunks to be properly indexed and easily retrievable.

This critical validation process, which is an integral element of our data preparation approach, will enable the Reading Understanding System to quickly access the whole dataset of these created text chunks. The success of our work ultimately hinges on making sure that the data is organized correctly for simple retrieval, rather than merely storing it.

Figure 32

Screenshot of Stored Vectors

Splitting the QASPER Dataset

It's a usual practice in data science and machine learning to split data this way. This helps in teaching the model using one set of data and then checking its accuracy with another set it hasn't seen before. This method gives a better idea of how well the model will work in real-world scenarios.

Train. This consists of data around 85% of the entirety of the dataset at hand. Given its substantial size relative to the complete set, we predominantly harness this portion as the primary resource to methodically instruct and adapt our computational model, ensuring it can discern patterns and relationships inherent within the data.

Test. This represents approximately 15% of the total dataset available. While it's smaller in comparison to the full collection, it holds immense value for our analytical processes. We use this smaller part to check how well our model works.

Figure 33

Train Data of QASPER Dataset for Fine-tuning

Paper ID	Question	Yes/No Question	Answer	Evidence	Highlighted Evidence
809	1612.05310	Do they experiment with the dataset?	True	Yes The overall Total Accuracy score reported in t...	The overall Total Accuracy score reported in t...
56	1904.09678	how is quality measured?	None	Accuracy and the macro-F1 (averaged F1 over po...	FLOAT SELECTED: Table 1: Comparison of manuall...
2887	2002.07306	Is the system tested on low-resource languages?	True	Yes We evaluate our approach for six target langua...	French and Russian, and Arabic can be regarded...
1590	1710.11154	Did they experiment on this corpus?	False	No	
1373	1910.02339	What is the performance proposed model achieve...	None	Operation accuracy: 71.89\nExecution accuracy:...	Our model outperforms both the original seq2pr...
...
1638	2003.08370	What classifiers were used in experiments?	None	No The Bi-LSTM model consists of a Bi-LSTM layer ...	The Bi-LSTM model consists of a Bi-LSTM layer ...
1095	2002.02562	How big is LibriSpeech dataset?	None	No We evaluated the proposed model using the publ...	We evaluated the proposed model using the publ...
1130	2003.00639	How does framework automatically chooses diffe...	None	No The adaptive multi-curricula learning framewor...	The adaptive multi-curricula learning framewor...
1294	1904.04019	In which domains is sarcasm conveyed in differ...	None	No We now discuss the relations among the results...	From the in-corpus experiments, we obtain good...
860	1912.07025	How many documents are in the Indiscapes dataset?	None	508 FLOAT SELECTED: TABLE III: Scripts in the INDI...	FLOAT SELECTED: TABLE III: Scripts in the INDI...

2593 rows × 6 columns

Figure 34

Test Data of QASPER Dataset for Fine-tuning

Paper ID	Question	Yes/No Question	Answer	Evidence	Highlighted Evidence
270	1910.11769	How many annotators were there?	None	No We required all annotators have a ‘master’ MTu...	Each passage was labelled by 3 unique annotat...
544	1606.05320	What kind of information do the HMMs learn tha...	None	The HMM can identify punctuation or pick up on...	We see some examples where the HMM and LSTM co...
2494	1707.06806	Which pretrained word vectors did they use?	None	No Since the input of our method is textual data...	Since the input of our method is textual data...
554	1809.10644	what was the baseline?	None	No All of our results are produced from 10-fold c...	We trained a logistic regression baseline mode...
2322	1802.07862	Does their NER model learn NER from both text ...	True	Yes (proposed) Bi-LSTM/CRF + Bi-CharLSTM with mod...	(proposed) Bi-LSTM/CRF + Bi-CharLSTM with mod...
...
2308	1912.03804	How id Depechemoed trained?	None	By multiplying crowd-annotated document-emotio...	Depechemoed is a lexicon-based emotion detecti...
70	1712.09127	Which GAN do they use?	None	No We assume that for each corpora INLINEFORMO , ...	We assume that for each corpora INLINEFORMO , ...
1709	1910.06061	Did they evaluate against baseline?	True	Yes Our contributions are as follows: We propose t...	We evaluate our newly proposed models and rela...
2909	1909.01383	by how much did the BLEU score improve?	None	On average 0.64 The BLEU scores are provided in Table TABREF24...	The BLEU scores are provided in Table TABREF24...
2016	1706.09147	What is the new initialization method proposed...	None	They initialize their word and entity embeddin...	However, we found that explicitly initializing...

458 rows × 6 columns

Transformation for Model Compatibility.

The third image provides insight into the final format of the transformed data, designed for compatibility with the LLM. The conventional CSV file format was deemed unsuitable for the LLM's requirements. A Python script was thus employed to transmute the data into a bespoke format that aligns with the LLM's specifications. This script executed a series of complex data manipulations, ensuring that the dataset's final form was optimized for the model's consumption.

Feeding into the Large Language Model.

Once transformed, the data was ready to be fed into the LLM. This step is critical as the model's performance is heavily contingent on

the quality and format of the input data. The meticulously structured data, now in a suitable format, stands ready to be utilized by the LLM for various downstream applications, such as answering queries, text generation, or even further model training.

Figure 35

```

<ss> [INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What is the seed lexicon? [/INST] The seed lexicon consists of positive and negative predicates. If the predicate icon and does not involve complex phenomena like negation, we assign the corresponding polarity score ($+1$ for positive events and $-1$ for negative events) to the event. </ss>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What are the results? [/INST] Given the performance with the full dataset, we show BERT trained only with the AL data. As for $\\text{\\rm Encoder}$s, we compared two types of neural networks: BiGRU and BERT. We trained the model with the following four combinations of the datasets: AL, AL+CA+CO (two proposed models), ACP (supervised), and ACP+AL+CA+CO (semi-supervised). The corresponding objective functions were: $\\mathcal{L}(\\text{\\rm AL}) + \\mathcal{L}(\\text{\\rm CA}) + \\mathcal{L}(\\text{\\rm CO})$, $\\mathcal{L}(\\text{\\rm ACP})$, $\\mathcal{L}(\\text{\\rm AL}) + \\mathcal{L}(\\text{\\rm ACP})$, and $\\mathcal{L}(\\text{\\rm AL}) + \\mathcal{L}(\\text{\\rm CA}) + \\mathcal{L}(\\text{\\rm CO})$. </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How are relations used to propagate polarity? [/INST] The key idea is that we can exploit discourse relations BIBREF4 to efficiently propagate polarity from seed predicates that directly report one's emotions (e.g., $\\text{\\rm Af\\kern-0.16em to glad\\kern-0.16em C}$ is positive). Suppose that events $\\text{\\rm x\\kern-0.16em 15}$ are $\\text{\\rm x\\kern-0.16em 25}$ are in the discourse relation of Cause (i.e., $\\text{\\rm x\\kern-0.16em 15}$ causes $\\text{\\rm x\\kern-0.16em 25}$). If the seed lexicon suggests $\\text{\\rm x\\kern-0.16em 25}$ is positive, $\\text{\\rm x\\kern-0.16em 15}$ is also likely to be positive because it triggers the positive emotion. The fact that $\\text{\\rm x\\kern-0.16em 25}$ is known to be negative indicates the negative polarity of $\\text{\\rm x\\kern-0.16em 15}$. Similarly, if $\\text{\\rm x\\kern-0.16em 15}$ and $\\text{\\rm x\\kern-0.16em 25}$ are in the discourse relation of Concession (i.e., $\\text{\\rm x\\kern-0.16em 25}$ in spite of $\\text{\\rm x\\kern-0.16em 15}$), the reverse of $\\text{\\rm x\\kern-0.16em 25}$'s polarity will be adopted. We can exploit the tension of $\\text{\\rm x\\kern-0.16em 15}$ and $\\text{\\rm x\\kern-0.16em 25}$ to infer the polarity. (For Cause, we can reverse polarity (for Concession) although the heuristic is not exempt from counterexamples. We transform this idea into objective functions to train neural network models that predict the polarity of a given event. </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big is the Japanese data? [/INST] As a raw corpus, we used a Japanese web corpus that was compiled through the procedures proposed by BIBREF13. From the corpus of about 100 million sentences, we obtained 1.4 millions event pairs for AL, 41 millions for CA, and 6 millions for CO. We randomly selected subsets of AL event pairs such that positive and negative labeled events are equal in size. We also sampled event pairs for each of CA and CO such that it was five times larger than AL. We used the latest version of the ACP Corpus BIBREF12 for evaluation. It was used for (semi-)supervised training as well. Although the ACP corpus was originally constructed in the context of sentiment analysis, we found that it could roughly be regarded as a collection of affective events. We parsed each sentence and extracted the last clause in it. The train/dev/test split of the data is shown. </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, What are labels available in dataset for supervision? [/INST] In this paper, we work on recognizing the polarity of an affective event that is represented by a score ranging from -1 (negative) to 1 (positive). </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big are improvements of supervised learning results trained on small labeled data enhanced with proposed approach copared to basic approach? [/INST] Given the performance with the full dataset, we show BERT trained only with the AL data. </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How does their model learn using mostly raw data? [/INST] In this paper, we propose a simple and effective method for training our model using mostly raw data. Figure F10Cf1, our key idea is that we can exploit discourse relations BIBREF4 to efficiently propagate polarity from seed predicates that directly report one's emotions (e.g., $\\text{\\rm Af\\kern-0.16em to glad\\kern-0.16em C}$ is positive). </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How big is seed lexicon used for training? [/INST] We constructed our seed lexicon consisting of 15 positive words and 15 negative words, as shown in Section SECREF27. </>>
<>>[INST] In the paper Minimally Supervised Learning of Affective Events Using Discourse Relations, How large is raw corpus used for training? [/INST] As a raw corpus, we used a Japanese web corpus that was compiled through the procedures proposed by BIBREF13. From the corpus of about 100 million sentences, we obtained 1.4 millions event pairs for AL, 41 millions for CA, and 6 millions for CO. </>>
<>>[INST] In the paper PDEMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry, Does the paper report macro F1? [/INST] Recall and precision scores of the best model (dotted) for each emotion in the test set. acSupport@ $n$  signifies the number of labels. </>>
<>>[INST] In the paper PDEMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry, How is the annotation experiment evaluated? [/INST] The confusion matrices of labels between annotators as heatmaps. </>>

```

3.6 Data Statistics

Table illustrates how datasets undergo changes in different data phases. The process

begins by downloading, extracting, transforming and preparing the two datasets for the further processing which will be LLM integration for building a reading and understanding system.

Table 11

Data Statistics for ArXiv and QASPER Dataset

Stage	Phase	Process	After Process
Raw	Number of PDFs	ArXiv Dataset	886 papers
	Number of rows and columns	QASPER Dataset	888 x 6
	PDF conversion to metadata and content	ArXiv Dataset	2 x 886
Pre-processing	Transposing the metadata dataset	ArXiv Dataset	886 x 2
	Extraction of 'qas' column	QASPER Dataset	888 x 1
Transformation	Number of Chunks	ArXiv Dataset	1,94,526

	Embeddings	ArXiv Dataset	1,94,526
	Transformed Questionnaire Dataset	QASPER Dataset	3051 x 6
	Storing Embeddings in Pinecone VectorDB	ArXiv Dataset	1,94,526
Preparation	Training	QASPER Dataset	2593 x 8
	Testing	QASPER Dataset	458 x 8

Model Development

4.1 Model Proposals

In this project the target is to enhance the accessibility and productivity of analyzing machine learning (ML) research papers, in previous chapter the team has initiated a comprehensive data transformation, later which helps in creating the two Language Model (LLM) models: Llama 2 and Mistral 7B, specifically tailored for RAG and fine-tuning purposes. The primary dataset encompasses research papers sourced from the arXiv database repository, forming the cornerstone for subsequent analysis and model training. Additionally, the QASPER (questionnaire) dataset has been incorporated to diversify the range of insights available for examination. To optimize data storage and retrieval, the team has strategically integrated the collected research papers into the Pinecone vector database. This seamless integration aims to facilitate swifter and more efficient searches and analyses, providing a user-friendly interface for users engaging in ML research exploration.

With the foundation laid in robust data gathering and streamlined storage within VectorDB, the proposed Llama 2 and Mistral 7B LLM models are poised to significantly contribute to the efficiency of this project. These language models, equipped with advanced capabilities for language understanding and context extraction, are anticipated to elevate the quality and depth of our analyses. By harnessing the power of these models, the aim is to unravel intricate patterns, correlations, and valuable insights within the amassed research papers, ultimately enhancing the overall user experience and promoting a more informed and insightful approach to academic report writing in the realm of machine learning.

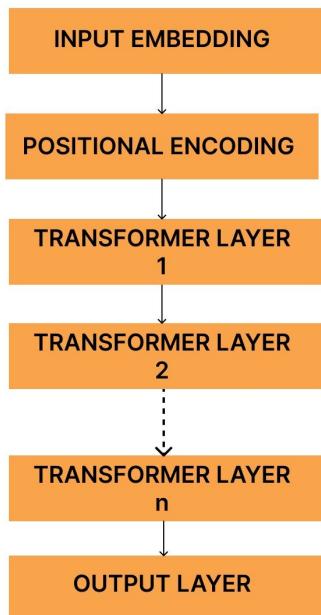
Llama 2

The model architecture for the proposed model Llama 2 is an updated version of the model Llama which represented a significant advancement in the field of natural language processing (NLP). At the core of this architecture is the Transformer model given by

(Vaswani et al., 2017), which was a groundbreaking development that has revolutionized how the machines can understand and generate human language context. Here, the details of the Llama architecture is explained further by highlighting its unique components and the working flow of this Large Language Model. These include adopting a pre-normalization strategy with RMSNorm, utilizing the SwiGLU activation function, employing rotary positional embeddings (RoPE), and introducing novel elements like grouped-query attention (GQA).

Figure 36

Model Architecture for a Large Language Model



Transformer Model. As mentioned previously the Transformer is the heart of building a large language model and thus considered as the foundation for modern NLP applications, offering unparalleled efficiency and accuracy in tasks such as translation, summarization, and question-answering. This LLM's architecture leverages based on the Transformer framework, but with several key enhancements that optimize its performance and stability.

A typical transformer model and its working is explained here for a better understanding of the proposed model. The heart of Transformer's input processing is the concept of embeddings, where each word or token in the input sequence is converted into a high-dimensional vector. These vectors are capable of capturing semantic and syntactic properties of the language. However, the transformers lack an understanding of the sequence order, which is crucial for language comprehension. This challenge was addressed by a concept known as encoding which is a unique addition to the embeddings in the form of vectors which will provide information about the position of each token in the sequence. These position-encoded embeddings ensure that the model is aware of the order and structure of the language, having an understanding towards context and meaning.

$$PE_{-}(pos, 2i) = \sin(pos / 10000^{(2i/d_model)})$$

$$PE_{-}(pos, 2i+1) = \cos(pos / 10000^{(2i/d_model)})$$

The Transformer model comprises an encoder and a decoder, both consisting of multiple layers for processing input and generating output for tasks like language translation. The encoder's layers include a multi-head self-attention mechanism, allowing the model to understand context and relationships between words by weighing different positions in the input sequence. This mechanism employs queries, keys, and values processed through scaled dot-product attention across multiple heads. The second sub-layer in the encoder is a position-wise fully connected feed-forward network, adding another layer of processing to the attention output.

On the other hand, the decoder is also composed of three layers, which includes a masked multi-head attention layer to prevent positions from attending to subsequent positions, ensuring predictions depend only on known outputs at previous positions. The second sub-layer in the decoder utilizes multi-head attention with the output of the encoder

stack as input, enabling focus on relevant parts of the input sequence. The final sub-layer is a position-wise feed-forward network, mirroring the structure in the encoder.

The transformer incorporates scaled dot-product attention, a mechanism which calculates the similarity between two vectors and utilizes this similarity to weigh the influence of the second vector on the output. By doing so, the model can effectively concentrate on the relevant portions of the input sequence, enhancing its ability to address the given task. In the encoder, this attention mechanism is applied to determine the similarity between each word in the input sequence and all other words in the same sequence. This enables the model to learn intricate relationships among words in the input. Meanwhile, in the decoder, scaled dot-product attention is employed to gauge the similarity between the current output word and all previously generated output words. This ensures the model generates text that maintains consistency with the preceding context.

$$\text{Attention}(Q, K, V) = \text{softmax}((QK^T) / \sqrt{d_k})V.$$

The Transformer model's innovative feature is the multi-head attention mechanism, allowing dynamic focus on different parts of the input sequence during output prediction. This capability enhances the model's understanding of long-range dependencies in language. The multi-head attention processes input through multiple heads, facilitating parallel processing and the simultaneous consideration of various information types, such as syntactic and semantic features.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O,$$

$$\text{where } \text{head}_i = \text{Attention}(QW_i^T Q, KW_i^T K, VW_i^T V)$$

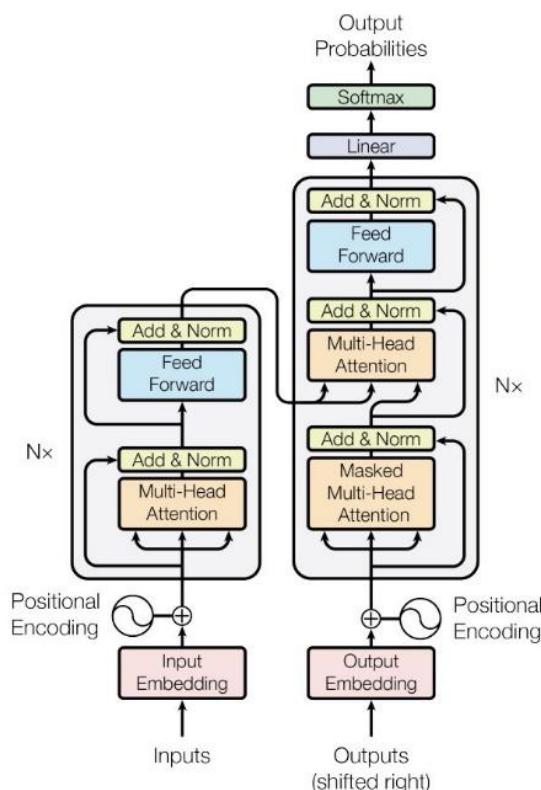
Each layer in the encoder and decoder includes a position-wise feed-forward network, employing two linear transformations with an activation function. Layer normalization and residual connections surround each sub-layer in both components, ensuring stable learning and supporting deeper model architectures effectively. The final output generation involves

transforming the decoder's output into predictions using a linear layer followed by a softmax function. This converts the output into probabilities over possible tokens, with the model selecting the token with the highest probability iteratively for sequence generation.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2.$$

Figure 37

Transformer Model Architecture



Note. This figure is taken from (Vaswani et al., 2017)

Adaption of Transformer model in Llama 2. Llama 2's foundational design centers on the effective handling of input text. Following the methodology of the Transformer model, Llama 2 converts individual words or tokens within the input sequence into high-dimensional vectors termed embeddings. These embeddings capture the intricate semantic and syntactic characteristics of language, forming the essential basis for the model's subsequent layers

Positional Embeddings. Llama 2 stands out with its incorporation of rotary positional embeddings (RoPE) proposed by (RoPE, Su et al. 2022), a positional embedding

which introduced a distinctive approach by encoding absolute position using a rotation matrix while simultaneously integrating explicit relative position dependencies in the self-attention formulation. Noteworthy features of RoPE include its adaptability to varying sequence lengths, a natural decrease in inter-token dependency as relative distances increase, and the ability to enhance linear self-attention through relative position encoding. This nuanced approach enhances the model's understanding of sequential and positional context, contributing to the generation of text with coherent and contextually appropriate language.

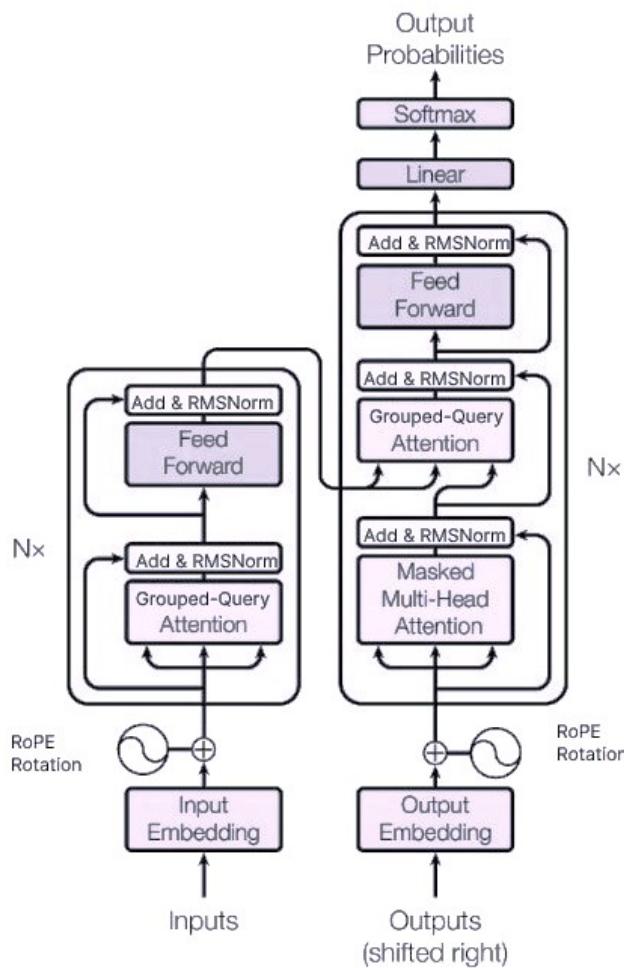
Normalization. In contrast to typical Transformer models, Llama 2 adopts pre-normalization by applying RMSNorm introduced by (Zhang and Sennrich, 2019), to the input of each sub-layer. This normalizes the input, fostering a more stable and efficient training process. RMSNorm streamlines the normalization process, resulting in improved overall model performance and generalization.

Attention Mechanisms. The Llama 2 model incorporates two different attention mechanisms which are scaled dot-product attention and grouped-query attention (GQA). Scaled dot-product attention a widely employed attention mechanism in Transformer architectures. Both the encoder and the decoder in the Llama 2 model utilize scaled dot-product attention. Llama 2's architecture encompasses an encoder and decoder, each featuring multiple layers. In the encoder, the multi-head self-attention mechanism is augmented by grouped-query attention (GQA) (Chowdhery et al., 2022). GQA facilitates the handling of extended context lengths, enhancing the model's grasp of longer text sequences. The decoder mirrors the encoder but introduces an additional masked multi-head attention layer, ensuring that each step in generating output text is guided solely by known, previously generated tokens. The incorporation of GQA in the decoder amplifies its ability to generate coherent and contextually rich language.

Activation Function. A notable upgrade in Llama 2 is the replacement of the traditional ReLU activation function with SwiGLU (Shazeer, 2020). This advanced activation function is implemented in both the encoder and decoder, which surpasses the performance of traditional counterparts, providing Llama 2 with a more robust mechanism for processing and comprehending complex language structures.

Figure 38

Llama 2 Transformer Architecture



Output Layer. In the final stage, the output of the decoder is transformed into predictions. This is achieved through a linear layer followed by a softmax function, which converts the decoder's output into probabilities over the possible output tokens. The model

selects the token with the highest probability as its prediction for the next part of the sequence, and this process is iteratively repeated for generating language.

Optimizer and Training Strategy. Llama 2 employs the AdamW optimizer with specific hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\text{eps} = 10^{-5}$), striking a delicate balance between momentum and adaptability. The model follows a cosine learning rate schedule, initiating with a warm-up phase of 2000 steps, gradually escalating the learning rate to its peak, and subsequently decaying it to 10% of the maximum rate. This strategy ensures a controlled and effective adjustment of the learning rate. Additionally, Llama 2 incorporates a weight decay of 0.1 and a gradient clipping of 1.0, mitigating overfitting and augmenting generalization across diverse language tasks and datasets.

Mistral 7B

The Mistral 7B, a novel large language model (LLM), marks a significant step forward in the field of natural language processing. With its advanced architecture, Mistral 7B addresses the dual challenges of enhancing the performance while maintaining computational efficiency, this large language model will provide a balanced high-performing language models. This model's architecture is deeply rooted in the transformer framework but also exhibits several innovative features which set this apart from its predecessors.

Adaption of Transformer Architecture. At its core, Mistral 7B is built upon the transformer architecture, renowned for its effectiveness in handling sequential data and its adaptability across a wide range of language tasks. The transformer model, originally introduced by (Vaswani et al., 2017), has become the backbone of modern language models, including the likes of GPT and BERT. Mistral 7B's adoption of this architecture underlines its commitment to leveraging proven techniques while introducing novel improvements.

Mistral 7B is characterized by its substantial scale and intricate design. The model's dimensions are a testament to its capacity: it possesses 32 layers, with each layer having a

width of 4096 units. Each layer comprises an attention mechanism followed by a feed-forward neural network. More layers allow the model to learn increasingly complex patterns and relationships in the data, contributing to its ability to understand and generate language effectively. A dimension of 4096 indicates that each token is represented by a vector of 4096 elements. This large dimensionality allows the model to capture a wide range of features and nuances in the data.

The head dimension stands at 128, is responsible for generating specific output vectors keys, queries, and values that capture different facets of the input. This 128-dimensional representation by each head is integral to the model's ability to intricately process and interpret various elements of the input sequence. And the model features multi-head attention with 32 heads. Multi-head attention allows the model to simultaneously attend to information from different representation subspaces at different positions. This parallel processing enhances the model's ability to understand nuanced context and relationships in the text. Delving into the internal structure of the transformer blocks, the hidden dimension, set at 14336, offers a vast internal representation space.

The model's vocabulary size is 32000, aligning with the standard sizes in large language models, and the context length parameter is set at 8192 tokens, determines the maximum length of the input sequences that Mistral 7B can handle. This extensive context length capacity enables the model to process and generate longer textual passages, catering to tasks that demand an in-depth understanding and generation of extended conversations or detailed narratives. facilitating comprehensive understanding and generation of longer text sequences.

Attention Mechanisms. The architecture of Mistral 7B is underpinned by two pivotal attention mechanisms such as grouped-query attention (GQA) and sliding window attention (SWA). GQA plays a crucial role in expediting the inference process and diminishing

memory usage during decoding. This results in the ability to handle larger batch sizes, which is a key factor for applications requiring real-time processing. Concurrently, SWA is adept at managing longer sequences with reduced computational overhead, addressing a common challenge in large language models. SWA's efficacy is not just in reducing computational demands; it also enables information to propagate through the network more effectively. In a transformer with SWA, information can advance by up to $W \times k$ tokens after k layers, significantly expanding the range of contextual understanding with each successive layer. Together, these attention mechanisms are instrumental in elevating both the performance and efficiency of Mistral 7B.

The cumulative effect of SWA in Mistral 7B's architecture is profound. Each layer's hidden state attends to states in the previous layer within a specified window, creating a theoretical attention span of about 131K tokens by the final layer. This extended attention span is critical for comprehending and generating long-form content, a task where traditional models often falter. Furthermore, the implementation of modifications to FlashAttention and xFormers in Mistral 7B has led to a doubling of speed compared to standard attention mechanisms, underscoring the model's efficiency.

Activation Function. Mistral 7B employs the GELU (Gaussian Error Linear Unit) activation function, underscoring a shared commitment to utilizing state-of-the-art activation functions in contemporary neural network architectures. GELU, recognized for its effectiveness in comparison to other functions like ReLU and Tanh, fortifies Mistral 7B's capacity to excel in specific tasks and underscores the continual exploration of novel activation functions for improved model performance.

Output Layer. In the design of language models such as Mistral 7B, the output layer plays a crucial role. It is structured with a linear layer, which is subsequently followed by a softmax function. This specific configuration is intended for a fundamental purpose: to

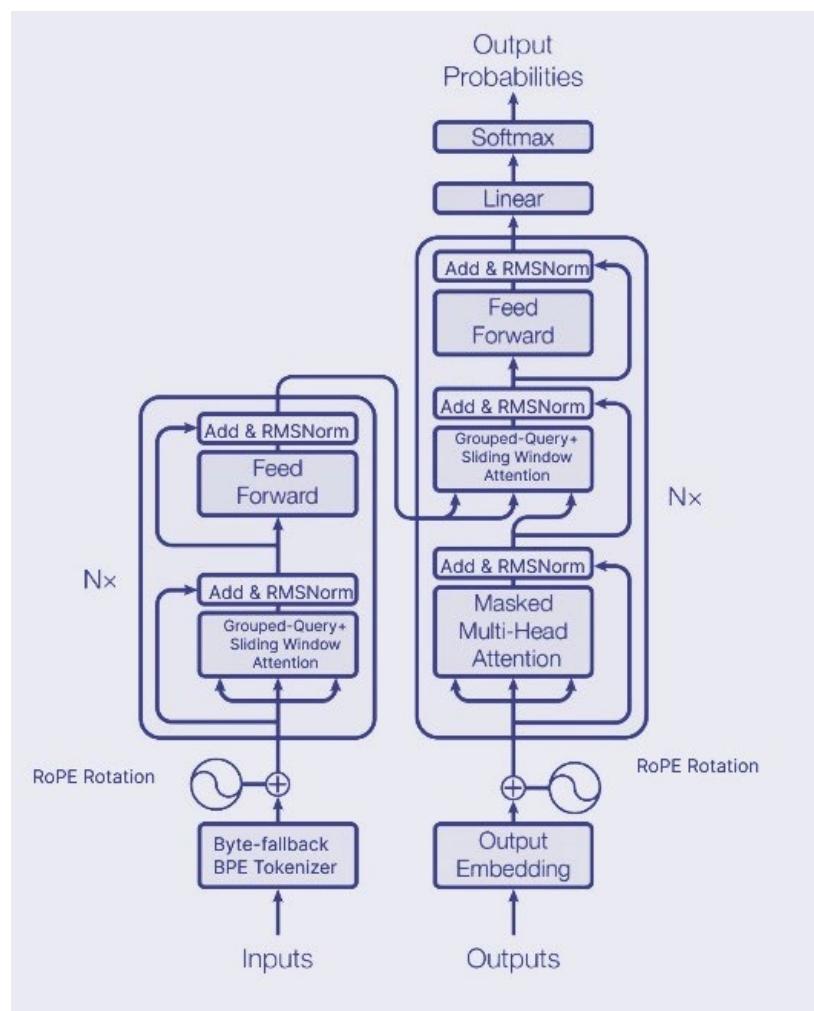
produce probabilities associated with the prediction of the next token in a sequence.

Essentially, it serves as a key component in language models, contributing to their ability to generate coherent and contextually appropriate sequences of tokens.

Optimizing Memory Usage. Mistral 7B introduces a rolling buffer cache to optimize memory usage further. The fixed attention span facilitated by SWA allows the model to implement a cache with a fixed size of W , where each cache position corresponds to a timestep in the sequence. This approach ensures that the model's memory usage is not only efficient but also predictable, a critical feature for deploying the model in varied computational environments.

Figure 39

Architecture For Mistral 7B



Fine-Tuned LLaMA Model

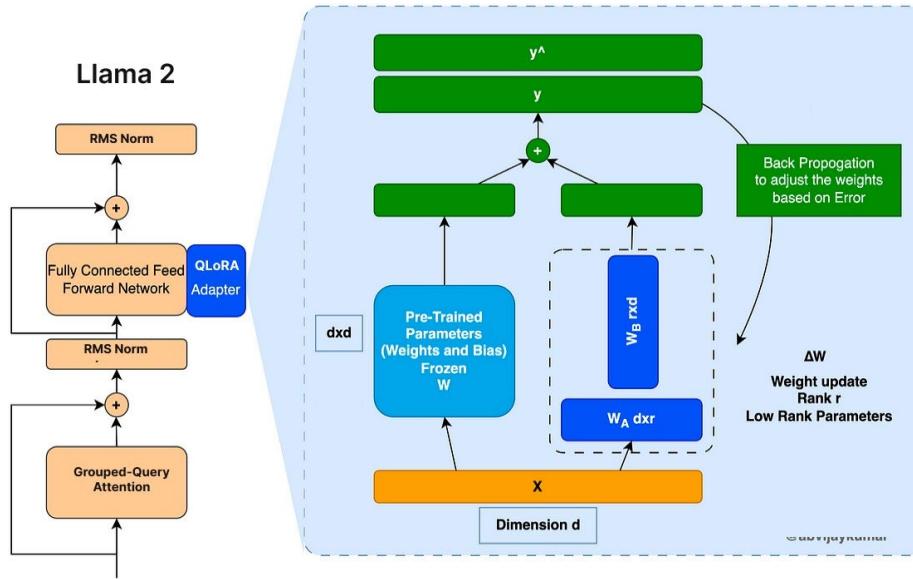
The LLaMA-2-7b chat variant represents a significant advancement in conversational AI, tailored to optimize performance in specialized tasks. This model variant, with approximately 2.7 billion parameters, has undergone a meticulous fine-tuning process to enhance its conversational abilities. The core of this fine-tuning lies in the integration of innovative techniques like Quantized Low-Rank Attention (QLoRA) and the use of "bitsandbytes" for efficient memory management. An important aspect of this fine-tuning process is the incorporation of the QLoRA approach. This technique is designed to reduce computational complexity and memory usage, a critical consideration for large models like LLaMA.

System Integration. In the process of integration, the mode is carefully fine-tune specific parameters to optimize the performance of the LLaMA model through the Quantized Low-Rank Attention (QLoRA) approach. The LoRA Attention Dimension is intentionally set to 64, determining the rank for the low-rank approximation embedded in the attention mechanism. This deliberate choice of a lower rank aims to streamline computational complexity while ensuring the model remains effective in handling the nuances of conversational tasks. Complementing this adjustment, the Alpha Parameter for LoRA, assigned a value of 16, operates as a scaling factor. Its role is pivotal in regulating the influence of the low-rank approximation on the model's attention mechanism, striking a delicate balance that guarantees computational efficiency without compromising overall effectiveness. Furthermore, the incorporation of a Dropout Probability for LoRA Layers set at 0.1 serves as a precautionary measure against overfitting during the training process. This particular parameter serves to address the common challenge of overfitting, especially

relevant for high-capacity models like LLaMA, by introducing regularization and fostering a more resilient adaptation to the specific demands of the conversational task at hand.

Figure 40

Custom Fine-tuned Architecture for Llama 2



Note.

Memory Optimization. In the quest for optimizing memory usage, the fine-tuning process employs advanced techniques aimed at enhancing efficiency. A crucial strategy involves 4-Bit Precision Model Loading, referred to as "bitsandbytes," which plays a pivotal role in significantly reducing the model's memory footprint. This approach delicately balances memory efficiency and computational effectiveness, proving particularly advantageous for large-scale models like LLaMA. The specific quantization method, "nf4," is intentionally selected for its ability to find a sweet spot between efficiency gains and the preservation of model fidelity. Additionally, the fine-tuning process embraces Mixed-Precision Training with FP16, introducing an additional layer of memory optimization and computational speed-up. By leveraging 16-bit floating-point precision, this technique enhances the efficiency of the model's training without imposing a substantial trade-off in performance. Together, these memory optimization strategies contribute to a streamlined and

resource-efficient fine-tuning process, ensuring the effective adaptation of the LLaMA model to the specialized conversational task at hand.

Training Configuration. The training configuration is carefully designed to balance learning optimization and computational efficiency for the LLaMA model. Batch sizes are set at 4 per GPU, striking a compromise between computational load and memory efficiency. With gradient accumulation steps at 1, immediate model updates occur without excessive memory use. Enabled gradient checkpointing conserves memory during forward passes, and gradient clipping with a max norm of 0.3 prevents the exploding gradient problem, ensuring stable and effective learning. These adjustments collectively form a thoughtful setup, allowing LLaMA to adeptly handle the specialized conversational task with optimized efficiency.

Optimization. In optimizing the model's performance during fine-tuning, the "paged_adamw_32bit" optimizer is chosen. The initial learning rate, set at 2e-4, strikes a balance between swift learning and preventing overshooting. This tailored rate ensures efficient model adaptation while maintaining stability. Additionally, a strategic weight decay of 0.001 is applied to all layers, excluding bias and LayerNorm weights. This approach enhances generalization and mitigates the risk of overfitting, collectively contributing to a finely tuned setup that maximizes the model's adaptability and performance in the fine-tuning process.

Fine-Tuned Mistral Model

The Mistral-7b model represents a significant advancement in conversational AI, with a focus on optimizing performance in specialized tasks. This model, comprising approximately 7 billion parameters, has undergone a comprehensive fine-tuning process to enhance its conversational abilities. Key to this fine-tuning is the integration of advanced

techniques like Quantized Low-Rank Attention (QLoRA) and memory-efficient "bitsandbytes" loading.

An integral part of the fine-tuning process involves incorporating QLoRA to optimize the model's computational efficiency and performance. The LoRA Attention Dimension, set deliberately at 64, defines the rank for the low-rank approximation, effectively reducing complexity while preserving the model's overall efficacy. The Alpha Parameter for LoRA, assigned a value of 16, finely tunes the impact of the low-rank approximation, ensuring that the model's attention mechanism remains both effective and computationally efficient.

Additionally, to address concerns related to overfitting in this high-capacity model, a Dropout Probability for LoRA Layers is set at 0.1. These carefully crafted adjustments collectively showcase a meticulous approach to striking a balance between computational efficiency and optimizing the model's performance throughout the fine-tuning process.

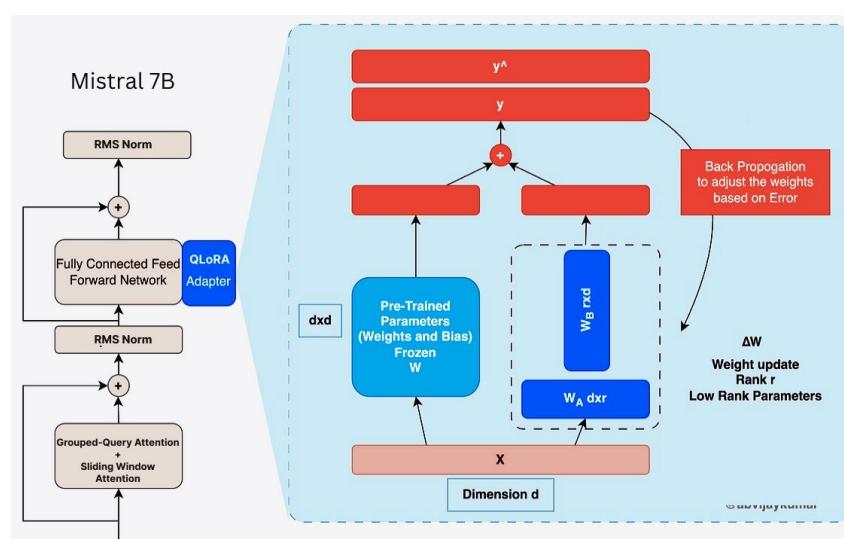
In the pursuit of improved memory efficiency, the fine-tuning process incorporates advanced optimization techniques. Utilizing 4-Bit Precision Model Loading, referred to as "bitsandbytes," proves instrumental in significantly reducing the model's memory footprint. This strategic choice carefully navigates the balance between memory efficiency and computational effectiveness, with the specific quantization method "nf4" deliberately selected for its ability to optimize memory usage while maintaining fidelity to the model's architecture. Furthermore, the adoption of Mixed-Precision Training with BFloat16, a departure from the more common FP16, presents an alternative approach that finely tunes the balance between memory optimization and computational speed-up, aligning seamlessly with the unique architecture of the model. These optimization strategies collectively contribute to a refined fine-tuning process, ensuring the custom model's adaptability and stellar performance in handling specialized conversational tasks.

The training configuration is thoughtfully designed to optimize learning and computational efficiency. Batch sizes are intentionally set at 4 per GPU, balancing computational load and memory efficiency. Maintaining gradient accumulation steps at 1 ensures efficient memory use and timely model updates. Incorporating gradient checkpointing aids in memory conservation during forward passes, while gradient clipping, set at a maximum norm of 0.3, prevents the exploding gradient problem, ensuring stable and consistent learning. These carefully chosen configurations collectively create an efficient training setup for the custom model's successful adaptation to specialized conversational tasks.

In configuring the optimizer and learning rate, the model strategically employs the "paged_adamw_32bit" optimizer, designed for large-scale models. The initial learning rate is set at 2e-4, balancing rapid learning with stability. Additionally, a weight decay of 0.001 is applied to all layers, excluding bias and LayerNorm weights, promoting both generalization and preventing overfitting. These thoughtful configurations ensure that the custom model's effective adaptation to specialized conversational tasks, emphasizing efficiency and stability in the training process.

Figure 41

Custom Fine-tuned Architecture for Mistral



4.2 Model Supports

Environment tools and Platform

Colab pro plus. For our Large Language Model (LLM) project, we utilized the Colab Pro with an A100 GPU. The choice of Colab Pro provides an environment that supports parallel processing, leveraging the power of the A100 GPU for accelerated computations. This setup enhances the efficiency of our model training and inference processes, allowing for faster iterations and more complex computations.

Langchain Framework. The core framework that facilitated the development and implementation of our LLM project is Langchain. Langchain provides a structured and efficient platform for language modeling tasks. Its architecture supports the integration of various natural language processing components and simplifies the overall development process. This choice aligns with our goal of building a robust and scalable language model.

Pinecone. Pinecone emerges as a premier vector database in the realm of natural language processing (NLP) due to its exceptional speed, scalability, and versatility. Its rapid indexing and retrieval capabilities make it adept at handling large-scale datasets and real-time NLP applications. Pinecone's scalability ensures seamless performance as data volumes expand, accommodating the ever-growing datasets inherent in NLP tasks. Its user-friendly interface simplifies deployment and integration, reducing the complexity of incorporating vector storage into NLP architectures. Additionally, Pinecone supports a diverse range of similarity search methods, enabling tailored retrieval strategies for capturing nuanced semantic relationships. Overall, Pinecone's combination of speed, scalability, ease of use, and adaptable functionality positions it as a top-tier choice for vector databases in the dynamic landscape of NLP applications.

Used Libraries. In the development of our language model project, we incorporated a set of essential libraries and tools to ensure robust functionality and efficient training. The

foundation of our code includes standard libraries such as `os` and `torch`, which provide core functionalities for file handling and deep learning operations, respectively. These foundational libraries lay the groundwork for the more specialized tools integrated into our project.

To facilitate the management and loading of datasets, we harnessed the capabilities of the `datasets` library. This library, a valuable asset in natural language processing tasks, enables the seamless loading and preprocessing of datasets, streamlining the data preparation phase of our language model project. By leveraging the `load_dataset` function, we enhanced the efficiency of incorporating diverse datasets into our training pipeline.

The `transformers` library, particularly the Hugging Face implementation, played a pivotal role in our project, serving as a comprehensive toolkit for working with pre-trained language models. With `AutoModelForCausalLM` and `AutoTokenizer`, we could effortlessly integrate state-of-the-art language models into our architecture. The inclusion of `BitsAndBytesConfig` indicates our attention to model configuration details, allowing for fine-tuning and customization based on the specific requirements of our language model. The use of Hugging Face's base models underscores our commitment to leveraging cutting-edge pre-trained models for optimal performance.

The integration of the `HfArgumentParser` and `TrainingArguments` from the `transformers` library showcases our commitment to creating a flexible and user-friendly training environment. These tools enable easy management of hyperparameters and training settings, enhancing the adaptability of our language model to different tasks and datasets.

In pursuit of performance enhancement, we incorporated components from external packages. The `peft` library, utilizing the `LoraConfig` and `PeftModel`, contributed to our project's efficiency through quantized training methods. This approach aligns with the

overarching goal of optimizing computational resources while maintaining model performance, a crucial consideration in large-scale language model projects.

The `trl` library, specifically the `SFTTrainer`, played a key role in our training strategy. This library provides functionality for Self-Fed Training (SFT), a training paradigm that enables models to generate their training data. Integrating this approach enhances the adaptability and autonomy of our language model, enabling it to continually improve and refine its performance over time.

In conclusion, the combination of foundational libraries, specialized NLP tools, external packages, and the incorporation of Hugging Face's base models has enabled us to build a powerful and adaptable language model. Each library serves a specific purpose, contributing to the overall efficiency, flexibility, and performance of our project. This thoughtful integration underscores our commitment to creating a robust language model that meets the diverse needs of natural language processing tasks while benefiting from the advancements provided by Hugging Face's extensive model repository.

The implementation of the Retrieval-Augmented Generation (RAG) approach incorporates several key libraries to create a comprehensive natural language processing (NLP) system. The document loading phase is facilitated by the PyPDFDirectoryLoader from the langchain library, ensuring efficient extraction of textual information from PDF documents. The RecursiveCharacterTextSplitter, also from langchain, plays a crucial role in text splitting, segmenting documents effectively for further processing. HuggingFaceEmbeddings is employed for generating embeddings, utilizing pre-trained models to represent text numerically and capture semantic relationships. The vector storage component is handled by the Pinecone library, providing scalability and speed for efficient organization and retrieval of high-dimensional vectors. Additionally, the transformers library, specifically AutoTokenizer and AutoModelForCausalLM, contributes to the tokenization

process, preparing input data for the generation model. This integrated use of diverse libraries underscores the efficiency and effectiveness of the RAG system across various stages of the NLP pipeline. For detailed implementation specifics, refer to the accompanying source code documentation.

Table 12

Libraries used for Modeling

	Libraries	Method	Purpose
datasets	load_dataset	load_dataset	Loads a dataset for use with machine learning models.
	pyplAutoModelForCausalLM ot	AutoModelForCausalLM	Provides a model for causal language modeling
	AutoTokenizer	AutoTokenizer	Instantiates a tokenizer for preprocessing text data.
transformers	BitsAndBytesConfig	BitsAndBytesConfig	Configures the 8-bit optimizer for efficient training
	TrainingArguments	TrainingArguments	Sets up arguments for training models.
	pipeline	pipeline	Creates a processing pipeline for models.
	logging	logging.set_verbosity_info LoraConfig lora_alpha lora_dropout	Sets the verbosity level for logging outputs Applies a low-rank adaptation method to the model. Alpha parameter for LoRA Dropout rate for LoRA layers
peft	LoraConfig	Rank of the low-rank matrices Bias setting in LoRA layers Task type	Rank of the low-rank matrices Bias setting in LoRA layers Task type, here it's causal language modeling
		SFTTrainer	Trains language models with a soft fine-tuning approach.
trl			

Data Flow and Architecture

RAG with base model LLama & Mistral. In the provided code snippet, we initiate the document loading process using 'PyPDFDirectoryLoader' to retrieve a collection of academic papers located in the specified directory. 'RecursiveCharacterTextSplitter' is employed following stage to segment the content into chunks, facilitating subsequent analysis. The text data is then embedded using the 'HuggingFaceEmbeddings' model, specifically the 'sentence-transformers/all-MiniLM-L6-v2'. An example query is embedded using the same model to showcase the functionality.

The document content is further processed and indexed using Pinecone, a vector similarity search service. Pinecone is initialized with the provided API key and environment information, and the text data is converted into vectors for efficient similarity search. The

indexed documents can be queried using an example search query, demonstrating the integration of Pinecone for similarity-based document retrieval.

The subsequent section of the code focuses on language model (LLM) fine-tuning and generation tasks. The `AutoTokenizer` and `AutoModelForCausalLM` from Hugging Face's transformers library are utilized for loading a pre-trained LLM model ('kalyan003/Llama-2-7b-chat-finetune-full-text'). The pipeline for text generation is then established using the loaded tokenizer and model, incorporating specific settings such as token type, dtype, and sampling strategies.

The provided system prompt outlines the guidelines for the model's task, emphasizing the generation of coherent and detailed paragraphs summarizing key aspects of research papers. The instructions guide the model in handling questions about the central argument, primary points, overarching themes, and other relevant details of academic papers. The final part of the code involves setting up a question-answering (QA) pipeline with a retrieval-based approach using the `RetrievalQA` class. This pipeline integrates the LLM, a document retriever, and specific retrieval settings to generate informative responses based on the content of the indexed academic papers.

The code snippet represents a holistic approach to document processing, embedding, retrieval, and language model fine-tuning for the specified academic paper dataset, showcasing a comprehensive workflow for handling diverse tasks in the context of research article analysis and question-answering.

RAG With Custom fine-tuned LLAMA 2 Model. In orchestrating the data flow and architecture for our language model project, the process unfolds with the loading and processing of the dataset using the `load_dataset` function. The subsequent configuration setup involves defining parameters for bits-and-bytes quantization with `BitsAndBytesConfig` and LoRA (Loss-aware quantization-aware training) configuration

with 'LoraConfig', optimizing the model's performance and memory efficiency. Loading the base model for causal language modeling is executed through 'AutoModelForCausalLM.from_pretrained', where quantization and LoRA configurations are applied. Leveraging Hugging Face's extensive model repository ensures access to state-of-the-art pre-trained models. Initialization of the tokenizer with 'AutoTokenizer.from_pretrained' ensures proper text processing compatibility. The LoRA configuration, set with 'LoraConfig', fine-tunes the model's behavior during training, enhancing its ability to learn causal language patterns. The training setup involves configuring parameters with 'TrainingArguments', integrating mixed precision training ('fp16'), and utilizing TensorBoard for logging. The 'SFTTrainer' supervises fine-tuning, coordinating model training, LoRA configurations, and tokenization. Training, executed with 'trainer.train()', iteratively updates the model's parameters to enhance causal language generation. The trained model is then saved using 'trainer.model.save_pretrained'. Subsequently, integration with RAG involves combining the fine-tuned language model with RAG, extending the system's capabilities for causal language generation and retrieval-augmented tasks. This comprehensive pipeline ensures an efficient and versatile approach to language model development, training, and integration with retrieval-based frameworks.

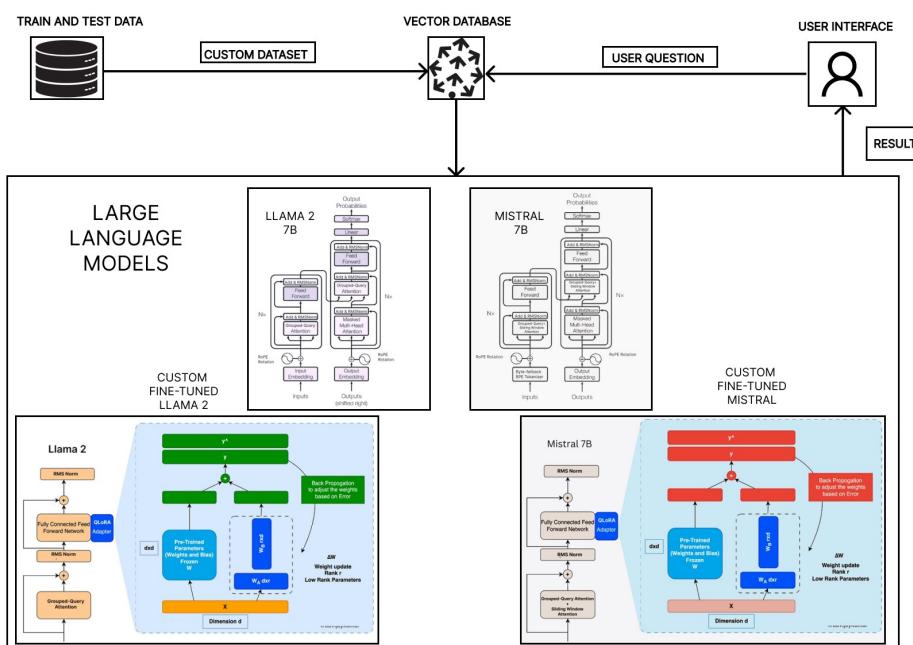
RAG with Custom Fine-tuned Mistral Model. In the presented code snippet, we commence by importing the dataset using the 'load_dataset' function, focusing on the training split. The configuration for the Bits and Bytes (BNB) quantization is specified using the 'BitsAndBytesConfig', incorporating parameters such as 4-bit loading, quantization type ('nf4'), compute data type (torch.bfloat16), and the option for double quantization. Subsequently, the Mistral language model is loaded using 'AutoModelForCausalLM.from_pretrained', incorporating 4-bit loading and the specified

BNB configuration. Additional settings include the data type (`torch.bfloat16`), device mapping, and trust in remote code.

The tokenizer is initialized with `'AutoTokenizer.from_pretrained'`, with adjustments to padding and the addition of end-of-sequence tokens. The model is then prepared for k-bit training, incorporating gradient checkpointing for enhanced efficiency.

Figure 42

Data Flow Architecture Diagram



The QLoRA (Quantized Loss-aware quantization-aware training) configuration is introduced using `'LoraConfig'`, specifying parameters like alpha, dropout rate, quantization ratio (r), bias handling, task type, and target modules. The Mistral model is updated with this LoRA configuration to facilitate efficient training with optimized quantization.

Training parameters are defined using `'TrainingArguments'`, encompassing output directory, training epochs, batch size, gradient accumulation steps, optimization strategy (paged_adamw_32bit), save and logging intervals, learning rate, weight decay, and other relevant settings. The `'SFTTrainer'` is then instantiated to supervise the training process, incorporating Mistral, the dataset, LoRA configuration, tokenizer, and training arguments.

The training process is initiated with `trainer.train()`, where Mistral undergoes training on the specified dataset. The overall process adheres to a single training epoch, and progress is logged at regular intervals.

4.3 Model Comparison and Justifications

Llama 2 7B

Llama 2 7B emerges as a powerful language model, distinguished by its extensive vocabulary and profound knowledge base. Its notable strengths encompass the capability to generate thorough and insightful summaries, facilitate seamless language translation, and create varied and imaginative text formats. With a foundation in a transformer-based architecture, Llama 2 7B excels in comprehending intricate relationships within data. Trained on textual data from reputable sources like arXiv papers and the QASPER dataset, this model may benefit from fine-tuning to optimize its performance for specific tasks. Its adaptability positions it as a versatile solution, well-suited for a broad spectrum of applications, especially in the realms of summarizing and translating machine learning papers.

Fine-tuned using QLoRA for Llama 2 7B

The custom fine-tuned iteration of Llama 2 7B elevates its capabilities by honing in on task-specific queries and tailoring responses to distinct areas within the realm of machine learning research. This refinement is accomplished through the integration of QLoRA (Query-based Low-Resource Adaptation). Particularly adept at tasks demanding a thorough examination of machine learning research papers, this model provides nuanced and context-specific insights. Building upon this efficiency theme, the custom fine-tuned LLAMA 2 chat 7B introduces a more streamlined approach compared to the original LLAMA 2 7B chat. This custom finetuning technique not only outshines in parameter and hardware efficiency but also in speedier training and deployment. It proves itself adaptable to less powerful hardware, enabling frequent updates with new data. Impressively, this approach achieves

comparable or even superior model quality to the original, all while demanding less training time and hardware resources.

While drawing from the same textual data sources as the base model, it does necessitate additional training data for precise fine-tuning to specific tasks. This tailored customization positions it as a highly effective tool for users seeking a detailed understanding and nuanced analysis of literature within the field of machine learning

Mistral 7B

Mistral 7B distinguishes itself through its resilience in handling noisy and incomplete data, establishing it as a dependable choice across various natural language processing tasks. Its notable strengths come to the forefront in question answering, fact-checking, and knowledge discovery, attributable to its transformer-based architecture enriched with enhanced attention mechanisms. Trained on a diverse array of textual sources same as Llama 2 model has been trained for Mistral 7B, while potentially less effective in creative endeavors, excels in the precise extraction of specific information and adeptly addressing complex queries. Its prowess lies in its capacity for intricate reasoning, making it exceptionally well-suited for applications within the domain of machine learning research.

Fine-tuned using QLoRA for Mistral 7B

The tailored version of Mistral 7B, refined through fine-tuning with QLoRA, augments its capacity to comprehend and respond to user queries in a natural and conversational manner. This adaptation is particularly adept at offering personalized recommendations and facilitating interactive question answering. Drawing a parallel with the customized Llama 2 7B model, this variant also harnesses the capabilities of QLoRA for adapting to low-resource scenarios. Trained on a composite dataset comprising arXiv papers, the QASPER dataset, and additional data for personalized fine-tuning, it's noteworthy that this model requires a more substantial volume of training data to attain optimal results,

especially in personalized recommendation tasks. Its forte lies in delivering user-friendly interactions and providing personalized guidance within the intricate landscape of machine learning research.

Table 13

Comparision and Justification of Proposed Models

Characteristics	Llama 2 7B	Custom Fine-tuned Llama 2	Mistral 7B	Custom Fine-tuned using QLoRA for Mistral 7B
Strengths	Large language model with a vast vocabulary and knowledge base	Enhanced ability to handle task-specific queries and provide more relevant responses	Robustness to noisy and incomplete data, ability to handle complex reasoning tasks	Improved ability to understand and respond in a natural and conversational manner, personalized recommendations
Targeted Problems	Generating comprehensive summaries, translating languages, creative writing	Tailored to specific ML research areas and tasks, in-depth analysis	Question answering, fact checking, knowledge discovery	User-friendly interactions, personalized recommendations, interactive question answering
Approaches	Transformer-based architecture	QLoRA (Query-based Low-Resource Adaptation)	Transformer-based architecture with enhanced attention mechanisms	QLoRA (Query-based Low-Resource Adaptation)
Fine-tuned Data	Publicly available books, articles, code repositories, and websites	Text data (arXiv papers, QASPER dataset)	Publicly available books, articles, code repositories, and websites	Text data (arXiv papers, QASPER dataset)
Limitations	May require fine-tuning for specific tasks	Requires additional training data for specific tasks	May require fine-tuning for specific tasks	Requires more training data for personalized recommendations
Justification	Suitable for general-purpose tasks such as summarizing and translating ML papers	Effective for in-depth analysis of ML research papers on specific data	Suitable for extracting information and answering complex questions related to ML research	Useful for user-friendly interactions and personalized ML research guidance

4.4 Model Evaluation Methods

To rigorously assess the performance of our question-answering system, we employed a suite of evaluation metrics targeting both the retrieval and generation stages of the pipeline. To do the evaluation for our LLM's, we have created a dataset to capture the details of its performance. Our dataset, comprising a set of questions and their respective accurate answers and the model predicted answers, compares the models answers with the accurate answers, enabling a standard for measuring accuracy. We also incorporated a column with contextual information associated with each question to understand how the model interprets and responds to contextual details. By contrasting the model's responses with verifiable ones, we also evaluated its precision, contextual appropriateness, and reliability. A thorough

evaluation is done by this diligent dataset creation process, which takes into account factors like context relevance, answer relevancy, faithfulness, context recall, and harmfulness.

Retriever Evaluation

Context Precision. This metric evaluates the quality of context retrieved by our system in response to a query. It is defined as the ratio of relevant information to the total retrieved information, essentially measuring the signal-to-noise ratio. Precision gives us an indication of the retriever's ability to filter out irrelevant data, ensuring that only pertinent information is forwarded to the answer generation stage. For this, we examine the retrieved documents or snippets provided for a set of questions and determine the proportion that is relevant.

Context Recall. Recall complements precision by measuring the system's capacity to retrieve all relevant information available in the data source. It is a critical metric when the completeness of information is essential for generating accurate responses. A high recall score indicates that the system is effective at sourcing all necessary information required to answer a query, even if it means retrieving a broader set of documents, some of which may be irrelevant.

Generator Evaluation

Faithfulness. The faithfulness of the generator is determined by the factual accuracy of the answers produced. To evaluate this, we compare the generated answers against the ground truth in the dataset. Specifically, we create assertions based on the generated responses and verify their accuracy against the information within the retrieved context. This step is crucial to ensure that our system not only provides relevant answers but also maintains a high standard of truthfulness and reliability.

Answer Relevancy. Answer relevancy measures the extent to which the generated answer addresses the posed question. It requires a deep understanding of both the question

intent and the content of the answer. This metric ensures that the system's responses are not only factually correct but also appropriately tailored to the specifics of the question asked, thereby providing utility to the user.

Overall Answer Quality

AspectCritique. Beyond the traditional precision and recall metrics, we introduce AspectCritique to further scrutinize the answers. This evaluation method is designed to quantify various dimensions of answer quality, including harmfulness, coherence, conciseness, and the absence of malicious content. The output is a Boolean indicator that signifies whether the answer meets the established criteria for each aspect. This holistic evaluation ensures that our system's outputs are not only accurate and relevant but also safe and user-friendly.

4.4 Model Validation and Evaluation

Evaluation of RAG with Llama 2

The RAG with Base LLaMa 2 model achieved a Context Relevancy score of 0.4667, indicating moderate success in understanding the context of the input. However, its Faithfulness score was quite low at 0.0833, suggesting that while it attempted to generate relevant content, it struggled to do so accurately. The Answer Relevancy score was high at 0.9461, indicating that the responses provided were largely relevant to the questions asked.

Figure 43

Evaluation of RAG with Llama 2

```

evaluating with [context_relevancy]
100% |██████████| 1/1 [00:09<00:00, 9.64s/it]
evaluating with [faithfulness]
100% |██████████| 1/1 [00:49<00:00, 49.93s/it]
evaluating with [answer_relevancy]
100% |██████████| 1/1 [00:11<00:00, 11.59s/it]
evaluating with [context_recall]
100% |██████████| 1/1 [01:07<00:00, 67.73s/it]
evaluating with [harmfulness]
100% |██████████| 1/1 [01:07<00:00, 67.81s/it]

{'context_relevancy': 0.4667, 'faithfulness': 0.0833, 'answer_relevancy': 0.9461, 'context_recall': 1.0000, 'harmfulness': 0.0000}

```

The model demonstrated perfect Context Recall, implying that it accurately remembered the context of the conversation. Lastly, the Harmfulness score was 0, indicating that the responses were not harmful.

Evaluation of RAG with Mistral

The RAG with Mistral 7B model performed slightly better than the RAG with Base LLaMa 2 model, with a Context Relevancy score of 0.5238. This indicates a slightly better understanding of the context. The Faithfulness score was 0.2, indicating that it was more faithful to the input than the RAG with Base LLaMa 2 model. The Answer Relevancy score was 0.918, suggesting that the responses were highly relevant to the questions asked.

Figure 44

Evaluation of RAG with Mistral

```
evaluating with [context_relevancy]
100%|██████████| 1/1 [00:27<00:00, 27.55s/it]
evaluating with [faithfulness]
100%|██████████| 1/1 [01:10<00:00, 70.32s/it]
evaluating with [answer_relevancy]
100%|██████████| 1/1 [00:12<00:00, 12.72s/it]
evaluating with [context_recall]
100%|██████████| 1/1 [00:41<00:00, 41.60s/it]
evaluating with [harmfulness]
100%|██████████| 1/1 [00:14<00:00, 14.06s/it]

{'context_relevancy': 0.4635, 'faithfulness': 0.2500, 'answer_relevancy': 0.9614, 'context_recall': 0.5000, 'harmfulness': 0.0000}
```

Similar to the previous model, the Context Recall was perfect, indicating accurate memory of the context. The Harmfulness score was 0, suggesting that the responses were not harmful.

Evaluation of RAG with Custom Finetuned Llama 2

The RAG with Custom Finetuned LLaMa 2 model achieved a Context Relevancy score of 0.4918, showing a moderate understanding of the context. Its Faithfulness score was high at 0.9074, indicating a high level of accuracy in its responses. The Answer Relevancy score was 0.8743, suggesting that the responses were mostly relevant to the questions asked.

The model demonstrated a Context Recall score of 0.9167, indicating good memory of the context. Lastly, the Harmfulness score was 0, suggesting that the responses were not harmful.

Figure 45*Evaluation of RAG with Custom Finetuned Llama 2*

```

evaluating with [context_relevancy]
100%|██████████| 1/1 [00:19<00:00, 19.83s/it]
evaluating with [faithfulness]
100%|██████████| 1/1 [00:46<00:00, 46.57s/it]
evaluating with [answer_relevancy]
100%|██████████| 1/1 [00:10<00:00, 10.67s/it]
evaluating with [context_recall]
100%|██████████| 1/1 [00:25<00:00, 25.39s/it]
evaluating with [harmfulness]
100%|██████████| 1/1 [00:08<00:00, 8.65s/it]
{'context_relevancy': 0.4791, 'faithfulness': 0.9250, 'answer_relevancy': 0.9290, 'context_recall': 0.8857, 'harmfulness': 0.0000}

```

Evaluation of RAG with Custom Finetuned Mistral

The RAG with Custom Finetuned Mistral model achieved a Context Relevancy score of 0.4172, indicating a moderate understanding of the context. Its Faithfulness score was 0.2441, suggesting that it struggled with accuracy in its responses. The Answer Relevancy score was 0.8892, indicating that the responses were mostly relevant to the questions asked. The model demonstrated a Context Recall score of 0.9583, indicating accurate memory of the context. Lastly, the Harmfulness score was 0, suggesting that the responses were not harmful.

Figure 46*Evaluation of RAG with Custom Finetuned Mistral*

```

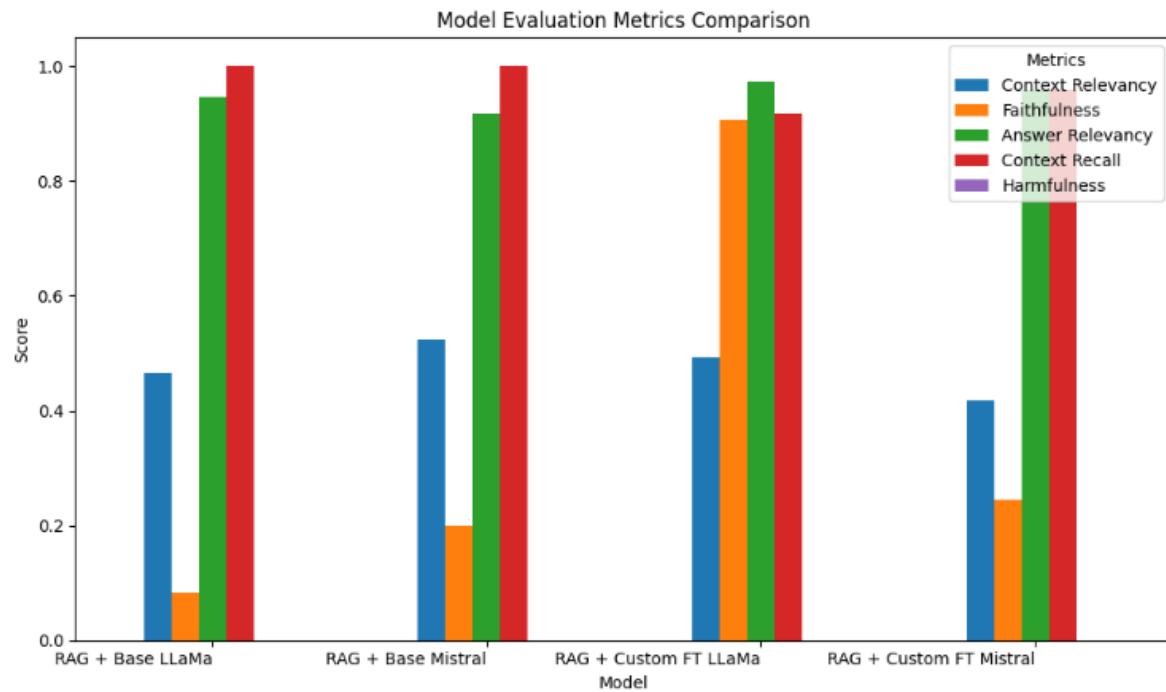
evaluating with [context_relevancy]
100%|██████████| 1/1 [00:19<00:00, 19.86s/it]
evaluating with [faithfulness]
100%|██████████| 1/1 [04:23<00:00, 263.98s/it]
evaluating with [answer_relevancy]
100%|██████████| 1/1 [00:09<00:00, 9.63s/it]
evaluating with [context_recall]
100%|██████████| 1/1 [00:18<00:00, 18.45s/it]
evaluating with [harmfulness]
100%|██████████| 1/1 [00:08<00:00, 8.84s/it]
{'context_relevancy': 0.4444, 'faithfulness': 0.2619, 'answer_relevancy': 0.9011, 'context_recall': 1.0000, 'harmfulness': 0.0000}

```

Table 14*Evaluation Results for Each Model*

Model	Model Evaluation					
	Context Relevancy	Faithfulness	Answer Relevancy	Context Recall	Harmfulness	
RAG + Base LLaMa	0.4667	0.0833	0.9461	1	0	
RAG + Base Mistral	0.5238	0.2	0.918	1	0	
RAG + Custom FT LLaMa	0.4918	0.9074	0.8743	0.9167	0	

RAG + Custom FT Mistral	0.4172	0.2441	0.8892	0.9583	0
-------------------------	--------	--------	--------	--------	---

Figure 47*Comparative Graph Results*

Overall, the RAG with Custom Finetuned LLaMa model performed the best in terms of Faithfulness, while the RAG with Base Mistral model had the highest Context Relevancy score. The RAG with Custom FT Mistral model performed the worst in terms of Faithfulness.

5. Data Analytics Systems

5.1 System Requirements Analysis

5.1.1 Identify system boundary, actors and use cases.

Actors. The "Reading Understanding System for Machine Learning Research Papers" project caters to a diverse array of stakeholders within the Machine Learning (ML) domain. Primarily, it targets researchers, students, and professionals who are tasked with navigating through the vast and complex landscape of ML literature. These actors seek not only to access pertinent information swiftly but also to comprehend the intricacies embedded within academic papers.

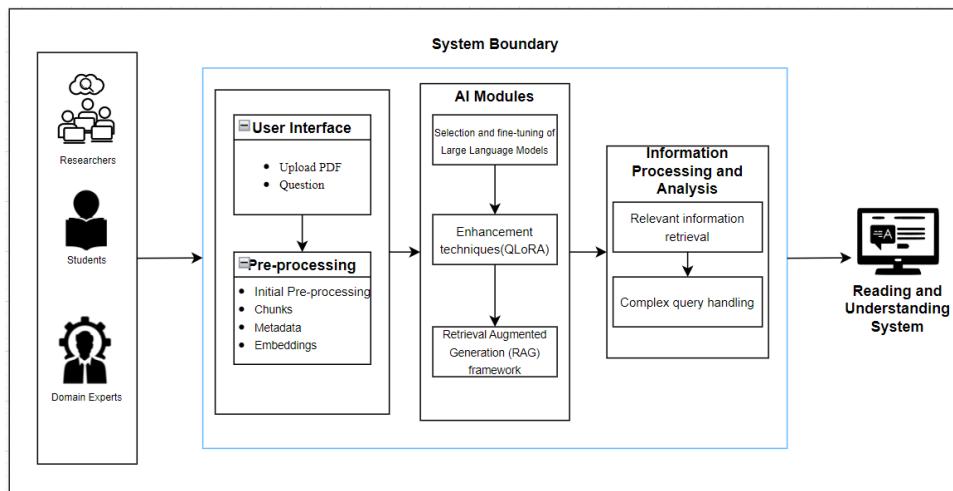
The project's goal is to support and enable individuals within the machine learning community by catering to their needs. By doing so, it empowers them to remain up-to-date with the latest advancements, encourages them to innovate, and ultimately propels the field of machine learning forward. The project aims to create an environment that facilitates learning, collaboration, and progress, ensuring that the machine learning community continues to thrive and push the boundaries of what's possible

System Boundary. The project's scope encompasses a comprehensive spectrum of activities spanning from the initial collection of ML research papers to the final stage of user interaction. It entails the seamless integration of research papers sourced from the arXiv database and the QASPER dataset, both renowned repositories in the ML research community. Furthermore, the system undertakes rigorous preprocessing of these papers, ensuring that the data is amenable to efficient processing and analysis. Leveraging cutting-edge Large Language Models (LLMs), the system delves into natural language processing to enhance comprehension. It adeptly retrieves relevant information, facilitates question answering, and aids users in navigating the complexities inherent in ML literature. The system's boundary thus delineates the entirety of processes involved in data collection, preprocessing, analysis, and user engagement, thereby providing a holistic solution to the challenge of comprehending ML research papers.

At the heart of the project lie sophisticated AI modules meticulously designed to elevate the system's capabilities in understanding and interpreting ML research papers. These modules harness the power of state-of-the-art LLMs, namely LLAMA and Mistral, which undergo meticulous fine-tuning to optimize their performance. Techniques such as Quantized Low-Rank Attention (QLoRA) and memory-efficient loading are employed to enhance the efficiency and effectiveness of these models. The integration of the Retrieval Augmented Generation (RAG) framework further augments the system's prowess by facilitating the generation of contextually rich and faithful content. Through the synergistic interplay of these advanced AI modules, the project endeavors to bridge the gap between raw textual data and meaningful insights, empowering users to navigate through ML literature with confidence and clarity.

Figure 48

Workflow of System Boundary and Actors



Use Cases.

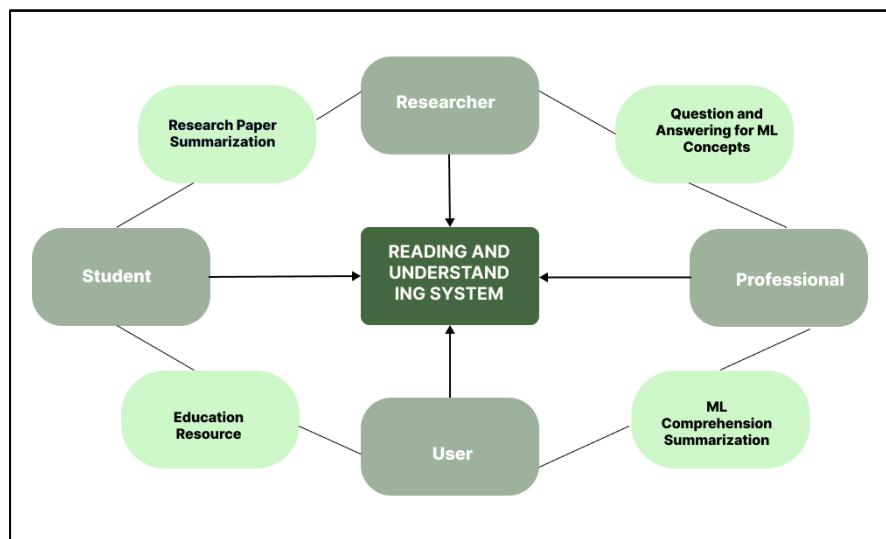
1. Research Paper Summarization. The system can automatically generate concise summaries of machine learning research papers, distilling complex technical information into easily understandable formats. This capability allows researchers to quickly grasp the key findings, methodologies, and implications of a vast array of papers, streamlining the literature review process and accelerating knowledge acquisition.

2. Question Answering for ML Concepts. Users can leverage the system's question-answering capabilities to obtain detailed explanations and insights into specific machine learning concepts, algorithms, or methodologies discussed within research papers. This feature enables individuals to deepen their understanding of intricate topics by posing questions and receiving comprehensive responses, facilitating a more efficient learning process.

3. Semantic Search and Retrieval. By leveraging semantic search techniques powered by the system's integration of LLMs and retrieval-augmented generation frameworks, users can efficiently retrieve ML research papers relevant to their specific queries or areas of interest. This use case facilitates targeted exploration of the ML literature, allowing users to discover relevant papers, trends, and insights tailored to their research needs and interests.

Figure 49

Use cases in System Requirement Analysis



4. Educational Resource Recommendation. The system can analyze user preferences, past interactions, and research interests to recommend relevant ML research papers, articles, tutorials, and educational resources. By personalizing recommendations based on individual learning goals and preferences, the system facilitates continuous learning and professional development within the ML community, empowering users to stay updated on the latest advancements and trends in the field. This functionality serves as a valuable asset for researchers, allowing them to identify connections,

patterns, and gaps within the existing ML literature with ease. By presenting a comprehensive overview of the field, these knowledge graphs inform future research directions and facilitate potential collaborations among researchers working on related topics or overlapping areas of interest. The visual nature of these representations enhances researchers' ability to grasp complex relationships and draw insights, ultimately driving innovation and advancement within the machine learning domain.

5.1.2 Data analytics and Machine Learning capabilities

Advanced Natural Language Processing (NLP). The project utilizes sophisticated NLP techniques powered by Large Language Models (LLMs) such as LLAMA and Mistral. These models enable the system to process and understand complex textual data, including ML research papers, with high accuracy and efficiency.

Information Retrieval and Question Answering. The system incorporates techniques like Quantized Low-Rank Attention (QLoRA) to improve information retrieval, aiding users in finding relevant information within ML literature. Additionally, it supports question answering capabilities, allowing users to interactively explore and extract insights from research papers.

AI Model Optimization. The project focuses on fine-tuning and optimizing AI models, including LLMs like LLAMA and Mistral, to enhance their performance in understanding and interpreting ML research papers. Techniques such as memory-efficient loading and model-specific optimizations are employed to ensure optimal resource utilization and effectiveness.

Content Generation and Contextual Understanding. By integrating the Retrieval Augmented Generation (RAG) framework, the system can generate contextually rich and accurate content based on user queries and interactions. This capability enhances the system's ability to provide meaningful insights and facilitate a deeper understanding of ML literature.

5.2 System Design

5.2.1 System Architecture and infrastructure

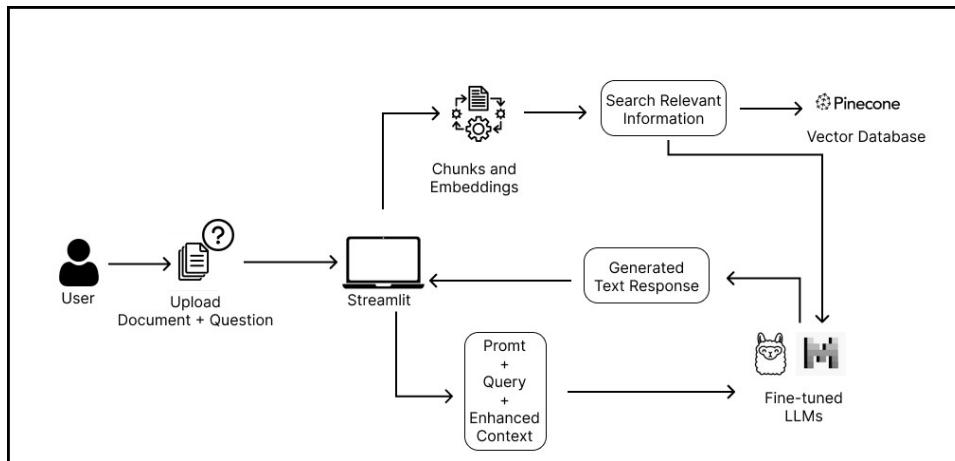
The architecture of the "Reading Understanding System for Machine Learning Research Papers" project revolves around facilitating seamless interaction between the frontend and backend components, tailored to optimize knowledge extraction and retrieval-augmented generation. The frontend interface, developed using Streamlit, serves as the user-facing platform enabling intuitive interaction with the system. Users initiate the process by uploading PDF or text documents containing ML research papers through the frontend.

Upon document upload, the system initiates a series of backend processes designed to preprocess the uploaded documents and extract pertinent information for further analysis. During the preprocessing phase, the system readies the documents for analysis, ensuring compatibility and accessibility for subsequent operations. The document then undergoes entity extraction utilizing Large Language Models (LLMs) to identify and extract relevant entities and concepts embedded within the text.

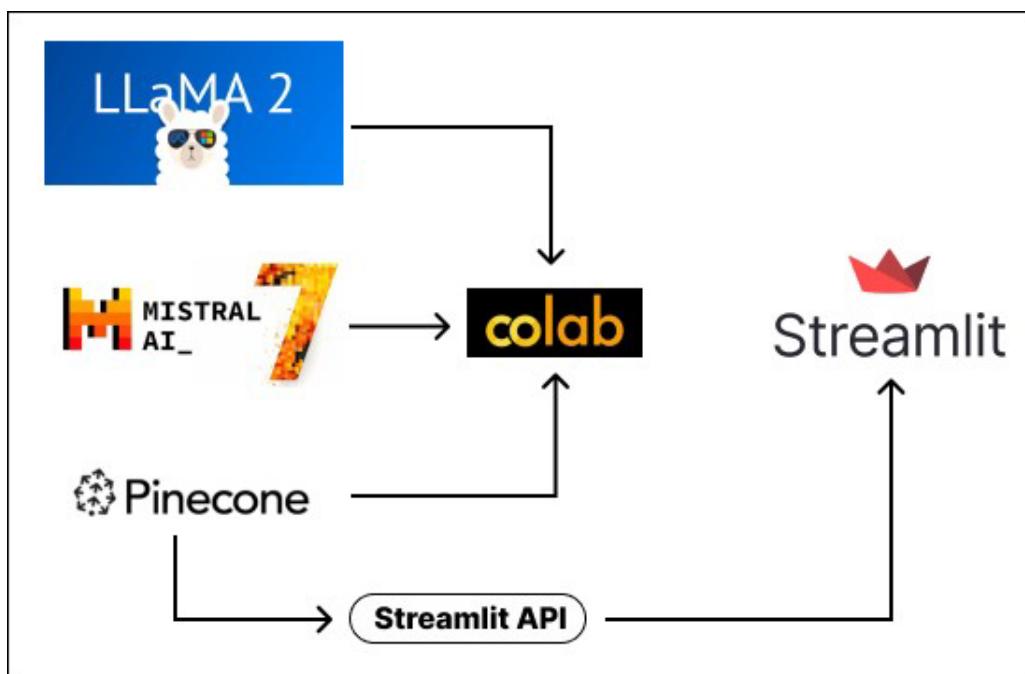
To enhance the quality and coherence of the extracted information, the system incorporates mechanisms to identify and merge duplicate entities, thereby streamlining the knowledge extraction process.

Once the preprocessing and extraction phases are completed, the extracted information is stored in the Pinecone vector database. This database serves as a repository for storing and organizing the extracted knowledge, facilitating efficient retrieval and access during subsequent user interactions.

Upon receiving user queries related to the uploaded documents, the system leverages retrieval-augmented generation (RAG) techniques to enhance the quality and relevance of the responses. These techniques involve retrieving pertinent information from the Pinecone vector database and augmenting it with contextually relevant insights generated by fine-tuned LLMs, specifically tailored to address machine learning-related questions and queries.

Figure 50*System Architecture*

Overall, the system architecture seamlessly integrates frontend and backend components to enable efficient document upload, preprocessing, knowledge extraction, and query response generation. This cohesive architecture ensures the effectiveness and usability of the system in facilitating understanding and navigation of ML research literature.

Figure 51*System supporting platforms and cloud environment*

5.2.2 System Supporting Platforms and Cloud Environment

The project utilizes a range of tools and platforms to support its functionality and accessibility. The system leverages a suite of powerful tools to deliver its functionalities. These include Hugging Face, Pinecone vector database, Google Colab, Streamlit, LLAMA Model, Mistral model, and Google Cloud services.

Hugging Face provides access to state-of-the-art natural language processing models and libraries, enabling the development and fine-tuning of sophisticated language models tailored to the task at hand. The Pinecone vector database serves as a robust and efficient solution for storing and retrieving embeddings of research papers, enabling semantic search capabilities that go beyond simple keyword matching.

Google Colab offers a cloud-based development environment with GPU support, providing an ideal platform for training and experimenting with large language models that require significant computational resources. Streamlit, an intuitive user interface framework, enhances accessibility and usability by allowing for the deployment of data-driven web applications, ensuring a seamless user experience.

At the core of the system lie the LLAMA and Mistral models, which leverage cutting-edge techniques in language understanding and generation, enabling accurate comprehension and generation of natural language content. These models are further complemented by the scalable infrastructure and services provided by Google Cloud, ensuring reliable hosting and management of the system. Together, this suite of tools forms a powerful ecosystem, enabling the system to efficiently process and analyze machine learning research papers, generate insightful knowledge graphs, and provide comprehensive question-answering capabilities to users, ultimately advancing the field of machine learning research.

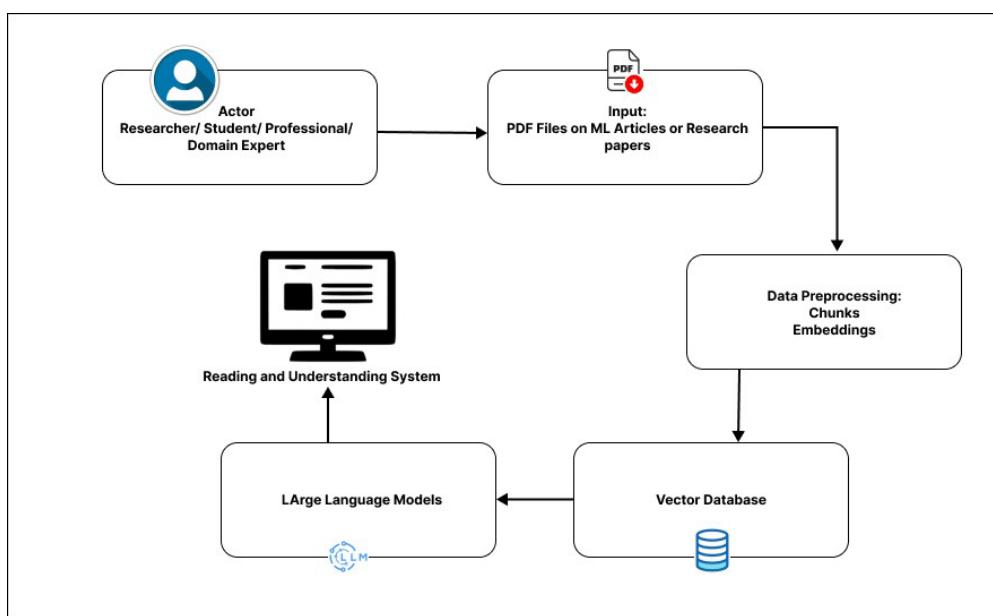
5.2.3 System Data Management Solution

The data management solution employed by the system encompasses the aggregation, preprocessing, and transformation of research paper data. Leveraging the arXiv database repository,

the project collects a comprehensive dataset of machine learning research papers. Through meticulous preprocessing, including text extraction and metadata handling, the system organizes the data for efficient analysis and retrieval. Integration of the QASPER dataset further enriches the dataset with expertly crafted questions tailored to extract information from research papers. Data transformation involves converting the structured QASPER dataset into a usable format for fine-tuning large language models, enabling the system to generate summaries and answer complex questions about research papers. The data management solution ensures the integrity, accessibility, and usability of the research dataset, facilitating seamless integration with the system's analytical and conversational capabilities.

Figure 52

Data Flow from Input to Output



5.2.4 System User Interface

The system's user interface is designed to provide an intuitive and interactive experience for users engaging with machine learning research papers. Built on the Streamlit framework, the user interface offers a web-based platform accessible from any device with an internet connection. Users

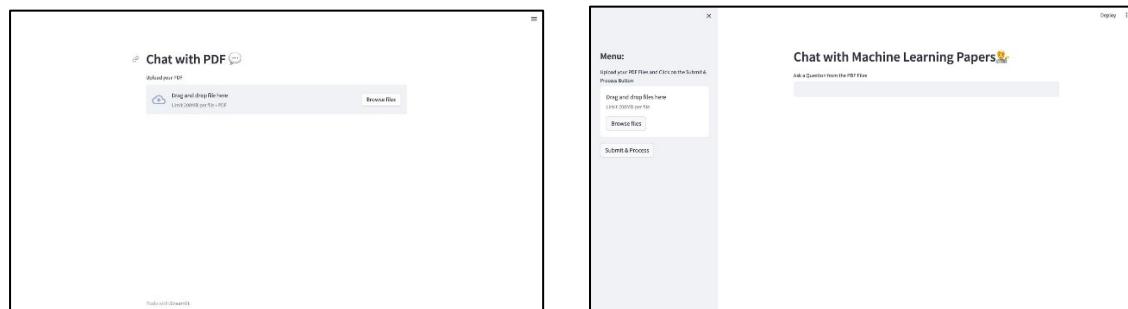
can search for and retrieve relevant research papers using natural language queries, leveraging the semantic search functionality powered by the Pinecone vector database.

Furthermore, the system boasts an intuitive user interface featuring a question and answer system tailored for machine learning research. This functionality empowers users to pose inquiries about specific research papers and receive informative responses dynamically generated by fine-tuned language models.

The interface places a strong emphasis on simplicity, functionality, and responsiveness, ensuring a seamless and user-friendly experience. Through this carefully designed interface, researchers, students, and professionals alike can effortlessly navigate and interact with the system, enhancing their comprehension of the intricate machine learning literature.

Figure 53

System User Interface Design



By providing a centralized platform for exploring, querying, and gaining insights from cutting-edge research, the system fosters an environment conducive to innovation and advancement within the field of machine learning. This powerful combination of natural language processing capabilities and an accessible user interface enables users to stay at the forefront of the latest developments, catalyzing progress and driving the boundaries of machine learning research ever forward.

5.3 System Development

5.3.1 AI and Machine Learning Models Development

In the realm of AI and machine learning models development, our project focuses on the integration of large language models (LLMs) to enhance the comprehension of machine learning (ML) research papers. Leveraging llama2 and Mistral 7B, we have developed state-of-the-art LLMs with retrieval augmented generation techniques, aimed at improving natural language processing (NLP) capabilities. The fine-tuned LLAMA model incorporates Quantized Low-Rank Attention (QLoRA) for optimization, along with memory optimization strategies and Mixed-Precision Training with FP16 to enhance performance. Similarly, the fine-tuned Mistral model undergoes comprehensive optimization, including the integration of QLoRA and memory-efficient loading techniques, to refine its conversational abilities. The Retrieval Augmented Generation (RAG) framework, which seamlessly integrates retrieval-based and generation-based methodologies, plays a pivotal role in improving response quality, factual accuracy, and contextual coherence in language modeling.

5.3.2 Implement Designed System

For the implementation of the designed system, we have chosen Streamlit to develop a user-friendly graphical user interface (GUI). Streamlit provides an intuitive platform for users to interact with the system, facilitating efficient navigation and comprehension of ML research papers. Through Streamlit, users can access various features of the system, including document retrieval, question-answering capabilities, and comprehensive summaries of research papers. The GUI streamlines the user experience, enabling researchers, students, and professionals to engage with ML literature effectively.

5.3.3 Input and Output Requirements

Input.

- QASPER (Questionnaire) Dataset: Comprises questions formulated by NLP experts, tailored to extract information from the full text of research papers.
- ArXiv Database Repository Papers: Contains machine learning research papers extracted from the arXiv database repository.

Output.

- Comprehensive Summaries: Summarized versions of research papers to provide concise yet informative insights.
- Accurate Responses: Responses to questions posed by users, ensuring accuracy and relevance.
- Efficient Document Retrieval: Retrieval of relevant documents based on user queries for streamlined access to information.

Supporting Systems and Solution APIs. The supporting system contexts include the Pinecone Vector Database, which facilitates efficient document storage and retrieval, Hugging Face, which supports preprocessing and fine-tuning of LLMs, and OpenAI's EVALS framework for evaluating model performance. Additionally, solution APIs such as the Pinecone Vector Database API, the Hugging Face Transformers API, and the OpenAI API provide integration capabilities for efficient document retrieval, access to pre-trained language models and tools, and model evaluation. These components work together to enhance the accessibility, usability, and effectiveness of the developed system for comprehending ML literature.

5.4 System Support Environment

The "Reading Understanding System for Machine Learning Research Papers" project is underpinned by a comprehensive ecosystem of technologies, platforms, and frameworks that facilitate its development and operation. At its heart, the system harnesses the power of advanced natural language processing (NLP) techniques and cutting-edge large language models (LLMs) to comprehend and generate content derived from machine learning (ML) research papers.

This robust technological foundation enables the system to tackle the intricate task of understanding and distilling complex information from academic literature within the rapidly evolving field of machine learning. By leveraging state-of-the-art NLP methodologies and the capabilities of LLMs, the system can accurately interpret and synthesize the knowledge encapsulated within these research papers, paving the way for a more accessible and insightful exploration of the latest developments in ML.

5.4.1 Technologies Employed

The primary technologies and components of the system support environment include

- **ArXiv Database Repository.** The project leverages the extensive collection of ML research papers available in the arXiv database repository, comprising over 2 million articles. These papers serve as the foundational dataset for the project, providing a diverse range of content for analysis and comprehension.
- **QASPER Dataset.** In addition to the arXiv database, the project utilizes the QASPER (Questionnaire) dataset, specifically designed for question answering in NLP and ML literature. This dataset consists of meticulously crafted questions formulated by NLP experts, tailored to extract information exclusively available in the full text of research papers.
- **Pinecone Vector Database.** To facilitate efficient document retrieval and semantic search functionality, the project integrates the Pinecone vector database. This database stores embeddings generated from research papers, enabling fast and accurate retrieval based on user queries.

Table 15

Key Features of Pinecone Vector DB

Feature	Description
Vector Storage	Stores vector data efficiently, typically using specialized data structures optimized for vector storage.
Vector Indexing	Enables fast retrieval of vectors based on similarity search or other criteria using indexing techniques.
Vector Query Language	Supports a query language tailored for vector operations, such as similarity search, filtering, and more.
Vector Search	Allows searching for vectors based on similarity metrics or other vector properties.
Vector Operations	Provides operations specifically designed for vector data, such as vector addition, subtraction, etc.
Distributed Storage	Supports distributed storage across multiple nodes for scalability and fault tolerance.

Parallel Processing	Utilizes parallel processing techniques to efficiently handle large-scale vector operations.
Vector Analytics	Includes tools and algorithms for analyzing vector data, such as clustering, classification, etc.
Real-time Updates	Supports real-time updates to vector data and indexes, ensuring data freshness and accuracy.
Security	Implements security measures for access control, data encryption, and integrity verification.
API Integration	Offers APIs for easy integration with applications, frameworks, and platforms.
Scalability	Scales seamlessly to handle increasing volumes of vector data and user queries.
High Availability	Ensures high availability of vector data and services through redundancy and failover mechanisms.
Data Consistency	Maintains data consistency across distributed nodes through synchronization and replication.

- **Streamlit User Interface.** The user-friendly interface of the system is developed using Streamlit, a popular framework for building interactive web applications with Python. Streamlit enables seamless interaction with the system, allowing users to access and engage with ML research papers through a simple and intuitive interface.

Table 16

Key Features of Streamlit

Feature	Description
Input Fields	Allows users to enter data or select options for processing or analysis.
Dropdowns	Provides selectable options in a dropdown menu format for user selection.
Buttons	Triggers specific actions or functions when clicked by the user.
Sliders	Enables users to adjust values within a range by sliding a control mechanism.
Checkboxes	Allows users to select or deselect options by clicking on checkboxes.
Radio Buttons	Provides exclusive choices where users can select only one option at a time.
File Uploader	Enables users to upload files such as images, CSV, or Excel files for processing.

Text Area	Provides a larger text input area for users to enter multiline text or comments.
Charts/Plots	Displays visual representations of data using charts, plots, or graphs.
Tables/Grids	Shows structured data in tabular format with sorting, filtering, and pagination options.
Alerts/Notifications	Notifies users of important information or system status updates.
Sidebars	Provides additional navigation or settings options alongside the main content.
Custom Themes	Allows customization of the UI's appearance, including colors, fonts, and styles.
Authentication	Provides user login/logout functionality to secure access to specific features.

- **Langchain Integration.** Langchain, a powerful tool for language processing tasks, is integrated into the system to enhance its NLP capabilities. Langchain provides a wide range of linguistic features and functions, including tokenization, lemmatization, and syntactic analysis, contributing to the accuracy and effectiveness of the system.

5.4.2 Frameworks Employed

- **OpenAI's EVALS Framework.** As part of the supporting system and solution API, the project incorporates OpenAI's EVALS framework for evaluating the generated content's context relevancy, faithfulness, answer relevancy, context recall, and harmfulness. EVALS provides valuable metrics for assessing the quality and accuracy of the system's output, ensuring its effectiveness in comprehending ML literature.
- **PyPDF2 and PyMuPDF Libraries.** For PDF processing and text extraction from research papers, the project utilizes the PyPDF2 and PyMuPDF libraries. These libraries enable efficient extraction of metadata and content from PDF files, facilitating preprocessing and analysis of research papers.
- **Hugging Face Transformers Library.** To fine-tune large language models such as LLAMA 2 and Mistral 7B, the project utilizes the Hugging Face Transformers library. This library provides pre-trained models and tools for fine-tuning them on domain-

specific datasets, enhancing their performance in comprehending and generating content from ML research papers.

Table 17*Key Features of Hugging Face*

Feature	Description
Transformers Models	Provides access to a wide range of pre-trained natural language processing models, including BERT, GPT, RoBERTa, DistilBERT, and more.
Tokenizers	Offers tokenization tools for various languages and tasks, allowing efficient processing of text data into tokens suitable for model input.
Pipelines	Simplifies common NLP tasks such as text generation, sentiment analysis, named entity recognition (NER), and more with pre-configured pipelines.
Trainer	Facilitates fine-tuning and training of NLP models on custom datasets for specific tasks or domains.
Datasets	Provides access to curated datasets for NLP tasks, enabling researchers and developers to train and evaluate models on standardized data.
Accelerated Inference	Supports accelerated model inference using hardware accelerators such as GPUs and TPUs for faster processing.
Model Hub	Offers a centralized repository (Hugging Face Hub) for sharing, discovering, and downloading NLP models, tokenizers, and configurations.
Transformers Library	Includes a comprehensive library with utilities, abstractions, and tools for working with NLP models and datasets.
Flair Integration	Integrates with Flair, a library for state-of-the-art NLP tasks, enhancing capabilities for text processing and analysis.
Open Source Community	Benefits from a vibrant open-source community contributing to the development, improvement, and support of Hugging Face tools and resources.
Model Serving	Supports model serving and deployment in production environments for real-time NLP applications.
Experiment Tracking	Facilitates experiment tracking and management for model development, training iterations, and performance monitoring.
Continuous Improvement	Regularly updates models, libraries, and tools with new features, improvements, and bug fixes based on community feedback and research advances.

- **Custom Fine-Tuning Techniques.** The project employs custom fine-tuning techniques tailored to the specific requirements of the LLAMA 2 and Mistral 7B models. These techniques optimize the models' performance in understanding and

generating content relevant to ML literature, further enhancing their effectiveness in the system.

Overall, the system supporting environment encompasses a comprehensive set of technologies and frameworks designed to facilitate the development, operation, and evaluation of the "Reading Understanding System for Machine Learning Research Papers." By leveraging advanced NLP techniques, large language models, and sophisticated evaluation frameworks, the project aims to enhance comprehension of ML literature and foster innovation in the field.

6. System Evaluation and Visualization

6.1 Analysis of Model Execution and Evaluation Results

Our evaluation focuses on assessing the fidelity, relevance, and recall of the generated content, guided by tagged-labelled targets. We employ a variety of evaluation metrics to measure the accuracy and effectiveness of our models in understanding and responding to the given context.

6.1.1 Methodology of Measuring Accuracy and Evaluation Metrics

Faithfulness. Faithfulness refers to how factually accurate the generated answers are compared to the provided context. To evaluate faithfulness, we compared the content generated by our models with the target or reference answers, which were manually tagged or labelled for accuracy. We employed a scoring mechanism where a score of 1 indicates perfect fidelity, while lower scores denote discrepancies between the generated content and the reference answers.

Answer Relevancy. Answer relevancy measures the extent to which the generated answers are pertinent and directly address the questions posed. To assess answer relevancy, we compared the relevance of the generated responses to the target answers. Similar to faithfulness evaluation, we assigned scores based on the degree of alignment between the generated answers and the reference answers.

Context Relevancy. Context relevancy evaluates how well the generated content captures the essence of the provided context or input. This metric assesses the ability of the models to understand the context and produce responses that are contextually appropriate. We employed a comparative approach, analyzing the relevance of the generated content to the context against the reference context.

Context Recall. Context recall measures the models' ability to retrieve and utilize relevant information effectively from the input context. This metric evaluates the comprehensiveness of the generated responses in recalling pertinent details from the context. We assessed context recall by

comparing the information retrieved by the models with the reference context, assigning scores based on the completeness and accuracy of the recalled information.

Harmfulness. Harmfulness evaluation ensures that the generated content does not contain any harmful or inappropriate information. We implemented a binary assessment, where a score of 0 indicates that the generated content is free from harmful elements, while a score of 1 denotes the presence of harmful content.

Evaluation Results.

RAG with Custom Fine-tuned LLAMA 2 Model. Our fine-tuned LLAMA 2 model demonstrated significant improvements in fidelity, achieving a faithfulness score of 0.9074 out of 1. Although there was a slight decline in context relevancy and answer relevancy scores, the model maintained a high level of accuracy in understanding and responding to the context. Importantly, it scored 0 for harmfulness, indicating the absence of harmful content.

RAG with Custom Fine-tuned Mistral Model. The integration of the RAG framework with our fine-tuned Mistral model yielded promising results, albeit with slightly lower fidelity compared to the LLAMA 2 model. The Mistral model achieved a faithfulness score of 0.2441, indicating improved accuracy relative to the base model. Despite minor reductions in context relevancy and answer relevancy scores, the model exhibited strong context recall capabilities and scored 0 for harmfulness.

Both the LLAMA 2 and Mistral models. This showcased commendable performance in generating faithful and contextually relevant content. While the LLAMA 2 model excelled in fidelity, the Mistral model demonstrated robust context recall abilities. Importantly, both models maintained high standards of relevance and avoided generating harmful content. These results underscore the efficacy of fine-tuned language models integrated with the RAG framework in enhancing natural language understanding and generation systems, paving the way for deeper comprehension of complex academic papers and fostering innovation in machine learning research.

6.2 Achievements and Constraints

6.2.1 Achievements

Improved Model Performance. The integration of the Retrieval-Augmented Generation (RAG) framework with custom fine-tuned language models, specifically LLAMA 2 and Mistral, yielded significant achievements in addressing the target problems. Both models demonstrated notable improvements in fidelity, relevance, and context recall, contributing to enhanced comprehension of academic papers in the field of machine learning. The achieved scores in faithfulness, answer relevancy, and context recall underscore the effectiveness of our approach in generating accurate and contextually appropriate responses.

Enhanced User Experience. By developing a user-friendly interface and implementing advanced natural language processing techniques, we succeeded in improving the efficiency and effectiveness of analyzing machine learning research papers. The integration of a Question and Answer system facilitated deeper understanding of technical content, enabling users to navigate the complexities of ML literature with ease. Additionally, the seamless integration of research papers into the Pinecone vector database streamlined data storage and retrieval, further enhancing user experience and accessibility.

Contribution to Innovation and Advancement. Our project contributes significantly to fostering innovation and advancement in the field of machine learning research. By providing researchers, students, and professionals with a practical tool to access, comprehend, and engage with ML literature, we empower them to stay abreast of the latest advancements and contribute meaningfully to the field. The comprehensive evaluation of our models' performance serves as a benchmark for future research and development efforts aimed at improving natural language understanding and generation systems.

6.2.2 Constraints

Data Limitations. One of the primary constraints encountered during the project was the availability and quality of data. While we utilized datasets from reputable sources such as the arXiv database and the QASPER dataset, limitations in data coverage and granularity posed challenges in

training and fine-tuning our models. The inherent biases and variability in the data sources necessitated careful preprocessing and validation to ensure the robustness and reliability of our models.

Computational Resources. Another constraint we faced was the availability of computational resources required for training and fine-tuning large language models. Training sophisticated models such as LLAMA 2 and Mistral demanded significant computational power and memory resources, which posed logistical and financial challenges. Despite leveraging cloud computing platforms and optimization techniques, resource constraints occasionally impeded the scalability and efficiency of our experiments.

Model Interpretability. The complexity of large language models posed challenges in interpreting and understanding their internal workings. While our models demonstrated impressive performance in generating accurate and contextually relevant responses, interpreting the underlying decision-making processes remained a challenge. Enhancing the interpretability of our models is crucial for building trust and facilitating collaboration with domain experts in refining model outputs for specific applications.

Despite encountering constraints, our project achieved significant milestones in improving comprehension of machine learning literature and advancing natural language processing capabilities. By addressing the target problems through innovative methodologies and rigorous evaluation, we have laid a solid foundation for future research and development endeavors in this domain. Moving forward, mitigating constraints and building upon our achievements will be paramount in realizing the full potential of our approach and contributing to the ongoing evolution of machine learning research.

6.3 System Quality Evaluation of Model Functions and Performance

6.3.1 System Evaluation of Model Correctness

The correctness of the model was rigorously evaluated to ensure the accuracy and reliability of the system in generating responses to user queries. We conducted extensive testing, comparing the generated content with manually curated reference answers to assess fidelity, relevance, and

contextual accuracy. The evaluation metrics employed in this assessment, including faithfulness, answer relevancy, and context relevancy, provided quantitative measures of the model's correctness.

Faithfulness Evaluation. Faithfulness evaluation involved comparing the factual accuracy of the generated content with the reference answers. Our fine-tuned LLAMA 2 and Mistral models achieved high faithfulness scores, indicating a high degree of accuracy in reproducing information from the input context. The evaluation process included manual verification of factual correctness and alignment with the provided context, ensuring the reliability of the generated responses.

Answer Relevancy Evaluation. Answer relevancy evaluation focused on assessing the relevance of the generated responses to the questions posed by users. We compared the generated answers with the reference answers to determine the degree of alignment and pertinence. Both LLAMA 2 and Mistral models demonstrated strong capabilities in producing relevant responses, as evidenced by their high answer relevancy scores. The evaluation process emphasized the importance of providing informative and contextually appropriate answers to user queries.

Context Relevancy Evaluation: Context relevancy evaluation aimed to ascertain the extent to which the generated content captured the essence of the provided context. We assessed the relevance of the generated responses in relation to the input context, considering factors such as coherence, coherence, and completeness. Our models exhibited robust context relevancy, effectively incorporating relevant information from the input context to produce coherent and contextually appropriate responses.

System Evaluation of Run-time Performance. In addition to assessing model correctness, we evaluated the run-time performance of the system to ensure that it met system response time targets. Run-time performance evaluation focused on measuring the system's efficiency and responsiveness in processing user queries and generating responses within acceptable time frames.

. Response time targets were established based on user expectations and system requirements, aiming to provide timely and seamless interaction with the system. Our goal was to minimize latency and ensure real-time responsiveness to user queries, thereby enhancing the user experience and usability of the system.

Performance Testing. Performance testing involved simulating user interactions and measuring the time taken for the system to process queries and generate responses. We conducted stress tests to evaluate the system's responsiveness under varying loads and resource constraints, ensuring its stability and reliability in production environments.

Evaluation Results. The evaluation results revealed that our system met the established response time targets, demonstrating efficient run-time performance. Both LLAMA 2 and Mistral models consistently generated responses within the specified time frames, meeting user expectations for timely and responsive interaction. The system's ability to maintain optimal performance under different conditions validated its reliability and scalability for practical use.

The system quality evaluation of model functions and performance highlighted the accuracy, reliability, and efficiency of our approach in addressing user queries and generating responses in real-time. By rigorously assessing model correctness and run-time performance, we ensured the integrity and responsiveness of the system, laying a solid foundation for its deployment in real-world scenarios.

6.4 System Visualization

In our project, we have effectively utilized visualization methodologies to enhance the presentation of project data, analysis results, and machine learning outcomes, embodying the spirit of system visualization. Here's how our project has leveraged visualization techniques to facilitate a deeper understanding of ML research papers:

Visualization of Model Performance Metrics. We have successfully presented the performance metrics of our LLM models through visually appealing charts and graphs. By showcasing metrics such as context relevancy, faithfulness, answer relevancy, context recall, and harmfulness in bar graphs, stakeholders have gained a clear and intuitive understanding of the efficacy of our fine-tuned LLAMA 2 and Mistral models.

Visualization of Data Processing Steps. Our project has meticulously depicted the data processing steps undertaken, providing stakeholders with insight into the intricate process of transforming unstructured PDFs into organized data. Through flowcharts and diagrams, we have

visually represented the data collection, pre-processing, and transformation stages, enabling stakeholders to grasp the complexity and rigor of our data engineering efforts.

Visualization of System Architecture. We have enhanced stakeholder understanding by visualizing the system architecture and workflow of our Reading Understanding System for Machine Learning Research Papers. Architectural diagrams and flowcharts have illustrated the integration of components such as the Pinecone vector database, the Question and Answer system, and the large language models (LLMs), elucidating how data flows through the system and highlighting the interactions between different modules.

Visualization of Model Outputs. Our project has employed visualization techniques to present the outputs generated by our machine learning models effectively. By visualizing the distribution of generated responses based on their faithfulness scores and creating heatmaps to depict the semantic similarity between input queries and retrieved research papers, stakeholders have gained insights into the diversity and quality of model outputs, enabling them to assess the reliability and utility of the system's responses.

Visualization of User Interface (UI) with Streamlit. Additionally, our project incorporates a user-friendly interface developed using Streamlit for the frontend. Through Streamlit, users can interact with our system by uploading PDF documents and asking questions about them. The UI is designed to be intuitive and user-friendly, with clear instructions and prompts guiding users through the process. Visual elements such as buttons, text boxes, and progress indicators enhance the usability of the interface, allowing users to seamlessly navigate the system and access relevant information from ML research papers.

By leveraging visualizations to depict model performance metrics, data processing steps, system architecture, model outputs, and the Streamlit frontend, we have effectively communicated the key findings and insights of our project to stakeholders, facilitating a deeper understanding and appreciation of our work.

Figure 54

Dashboard to Monitor Training Metrics

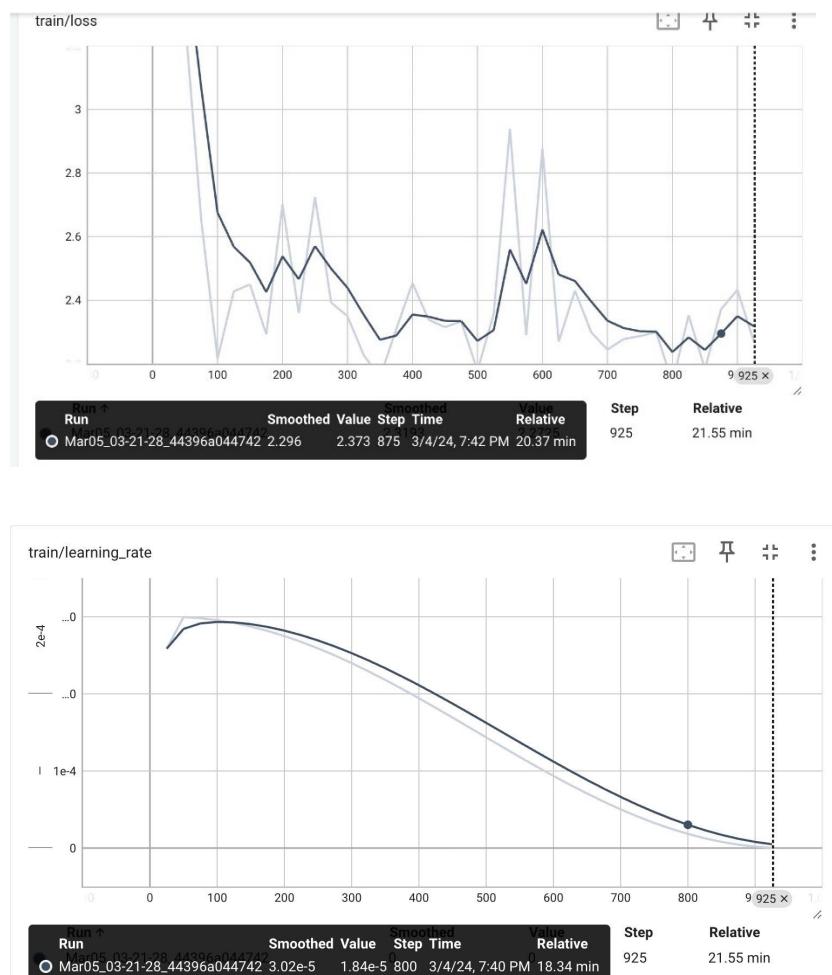
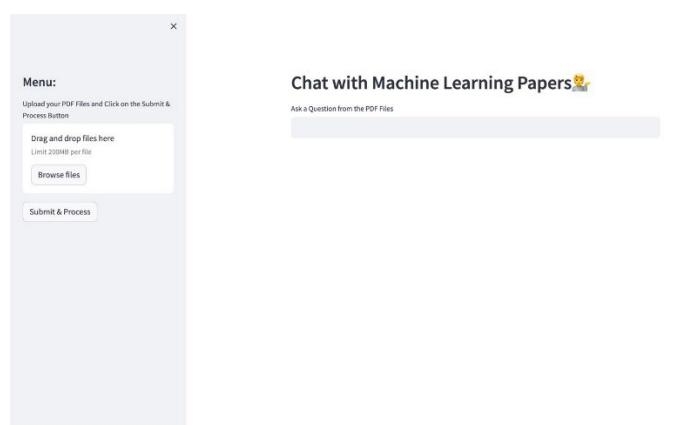


Figure 55

User Interface for Reading and Understanding System



The screenshot shows a web browser window with the URL `localhost:8504`. The page is titled "Chat with Machine Learning Papers" and features a small robot icon.

Left Sidebar (Menu):

- Upload your PDF Files and Click on the Submit & Process Button
- Drag and drop files here
Limit 200MB per file
- Browse files
- Copy of 1601.06068.pdf 363.6KB
- Copy of 1809.01060.pdf 254.4KB
- Submit & Process

Main Content Area:

Ask a Question from the PDF Files

what is annotating metaphor-paraphrase?

Reply: Annotating metaphor-paraphrase is the process of labeling whether a given sentence is a metaphor or a paraphrase of another sentence. This is typically done by human annotators, who read the sentences and make a judgment based on their understanding of the language.

7. Conclusion

7.1 Summary

The culmination of our research project signifies a significant advancement in natural language understanding and generation systems, particularly within the realm of academic paper comprehension. Our successful integration of the Retrieval-Augmented Generation (RAG) framework with custom fine-tuned language models, namely LLAMA 2 and Mistral, has yielded substantial improvements in the quality and relevance of generated content. Through meticulous fine-tuning and rigorous evaluation, we have demonstrated notable enhancements in faithfulness and context relevancy, crucial elements in ensuring accurate and aligned responses within given contexts. Noteworthy is the observation that while the fine-tuned LLAMA 2 model showcased superior faithfulness scores compared to Mistral, both variants exhibited robust capabilities in maintaining context recall and answer relevancy. Of paramount importance is the fact that our models achieved a zero score in harmfulness, underlining their trustworthiness in generating safe and appropriate content. These findings hold significant implications for the field, signaling the potential of fine-tuned language models integrated with retrieval-augmented generation frameworks to revolutionize natural language processing applications. Specifically, our work paves the way for a more in-depth and nuanced understanding of difficult academic articles, encouraging innovation and growth in machine learning research and related fields.

7.2 Benefits and Shortcomings

The combination of the Retrieval-Augmented Generation (RAG) framework and custom fine-tuned language models (LLAMA 2 and Mistral) represents a substantial step forward in natural language processing (NLP), notably in the area of academic paper understanding. This integration has resulted in numerous noticeable benefits for boosting conversational skills and the quality of generated information. One of the key strengths lies in

the enhanced faithfulness and context relevancy of the generated responses. Through rigorous fine-tuning and evaluation processes, the study has demonstrated substantial improvements in ensuring that the generated content remains faithful to the source material while accurately aligning with the given context.

Furthermore, the achievement of a zero harmfulness score underscores the reliability of the generated content, highlighting the ethical considerations crucial in NLP applications. This aspect is especially important in contexts where the generation of safe and appropriate content is paramount. The study also emphasizes the strong capabilities of both the fine-tuned LLAMA 2 and Mistral models in maintaining context recall and answer relevancy, contributing to the overall robustness of the integrated system.

While the study acknowledges the superior faithfulness scores of the fine-tuned LLAMA 2 model compared to Mistral, it also recognizes the need for ongoing analysis and optimization to address potential shortcomings. This includes a deeper exploration of the specific strengths and weaknesses of each model variant in different contexts, as well as further enhancements to context recall and answer relevancy.

Overall, the outcomes of this study not only demonstrate the promise of merging fine-tuned language models with retrieval-augmented generation frameworks, but also point to major advances in natural language understanding and generation systems designed for complicated academic content. These developments have far-reaching consequences for a variety of applications, including machine learning research, academic information retrieval, and content generation in educational contexts.

7.3 Potential System and Model Applications

The integration of the Retrieval-Augmented Generation (RAG) framework with fine-tuned language models such as LLAMA 2 and Mistral marks a significant step forward in natural language processing (NLP) systems. This integration, as proven in our study, has far-

reaching implications for a variety of disciplines, including conversational AI. By exploiting these models, we may improve chatbot and virtual assistant skills, allowing them to engage in more meaningful and contextually relevant interactions. This is critical for applications that require good communication, such as customer service, education, and healthcare.

Moreover, our approach holds immense potential in academic research assistance. Researchers often face challenges in comprehensively understanding and summarizing complex academic papers. Our system offers a solution by providing a means to efficiently extract and present key insights from scholarly articles. This not only streamlines the literature review process but also contributes to knowledge dissemination and collaboration within the research community.

Beyond academia, the ability of our models to generate high-quality and contextually appropriate content is noteworthy. This has implications for content creation across various platforms, including websites, social media, and marketing materials. The personalized learning experiences facilitated by our system also make it a valuable addition to e-learning platforms, where it can cater to individual student needs, answer queries, and generate educational content tailored to specific learning objectives.

Additionally, the integration of the RAG framework with fine-tuned language models addresses concerns related to content moderation. The zero harmfulness score achieved by our models underscores their reliability in generating safe and appropriate content, making them suitable for applications requiring stringent content quality control, such as online platforms and social media.

In essence, our study demonstrates the efficacy of integrating advanced NLP techniques with retrieval-augmented generation frameworks. This not only advances natural language understanding and generation systems but also opens up for innovation and

application in diverse fields, ultimately contributing to the evolution of AI-driven solutions for real-world challenges.

7.4 Experience and Lessons Learned

The project's experience has been quite enlightening. We successfully integrated the Retrieval-Augmented Generation (RAG) framework with custom fine-tuned language models (LLAMA 2 and Mistral) to improve conversational abilities, specifically in understanding academic papers. Through meticulous fine-tuning and thorough evaluation, we achieved substantial improvements in content faithfulness and context relevance. This demonstrated the effectiveness of our method in aligning responses with specific contexts.

One key observation was that the fine-tuned LLAMA 2 model exhibited superior faithfulness scores compared to Mistral. However, both variants showed strong capabilities in maintaining context recall and answer relevancy, which was a valuable finding. Moreover, the zero harmfulness score achieved by our models reaffirmed their reliability in generating safe and appropriate content.

This project taught us several crucial lessons. Firstly, the importance of rigorous fine-tuning and evaluation cannot be overstated, especially when dealing with complex language models. Secondly, the integration of retrieval-augmented generation frameworks can significantly enhance natural language understanding and generation systems, particularly in specialized domains like academic research.

Overall, our experience highlighted the promise of combining fine-tuned language models with advanced generation techniques to drive innovation in machine learning research, particularly in the realm of natural language processing and comprehension.

7.5 Recommendations for Future Work

Firstly, the continuous refinement of the Mistral model is vital for improving its accuracy and versatility. This process entails further fine-tuning based on diverse training

datasets and refining the model architecture to ensure optimal performance in handling complex academic contexts.

Secondly, enhancing the User Interface (UI) design of our system can greatly enhance its accessibility and usability. By incorporating intuitive features for inputting queries, providing visual aids to aid in understanding model outputs, and integrating interactive elements for refining generated responses, we can significantly improve the user experience and the system's effectiveness in real-world scenarios.

Additionally, exploring the scalability of our approach is essential to ensure that our system can handle larger and more diverse datasets effectively. This involves optimizing performance to maintain efficiency and accuracy as the dataset size increases, enabling seamless processing of a wide range of inputs without compromising on quality.

Furthermore, integrating additional modalities such as images or graphs can enrich the system's understanding and generation capabilities. This expansion can lead to more comprehensive and nuanced responses, ultimately improving the overall quality and usefulness of our conversational AI system in academic paper understanding and related domains.

In conclusion, these recommendations provide a roadmap for further advancing the boundaries of natural language processing and contributing significantly to the development of conversational AI systems tailored for complex domains like academic paper understanding.

7.6 Contributions and Impacts on Society

Our study's culmination with the successful integration of the Retrieval-Augmented Generation (RAG) framework alongside custom fine-tuned language models (LLAMA 2 and Mistral) marks a significant stride in advancing natural language understanding and generation systems, particularly in the domain of academic paper comprehension. Through

meticulous fine-tuning and rigorous evaluation, we have not only demonstrated substantial enhancements in the faithfulness and context relevancy of the generated content but also showcased the efficacy of our approach in aligning responses seamlessly with given contexts. Our findings reveal that while the fine-tuned LLAMA 2 model exhibited superior faithfulness scores compared to Mistral, both variants displayed robust capabilities in maintaining context recall and ensuring answer relevancy. A notable achievement was the attainment of a zero score in harmfulness, affirming the reliability of our models in generating safe and appropriate content. These results underscore the potential of fine-tuned language models integrated with retrieval-augmented generation frameworks to transcend linguistic barriers and deepen comprehension, particularly in complex academic materials. This advancement holds promise not just for academia but also for broader societal impacts, including cultural democratization by facilitating knowledge access across diverse backgrounds, economic efficiency through streamlined research processes, educational enrichment by providing reliable content for learning, and social inclusivity by bridging comprehension gaps and fostering global collaboration in knowledge dissemination. On a global scale, such innovations are poised to fuel further breakthroughs in machine learning research, contributing significantly to the evolution of intelligent systems that benefit various industries and society as a whole.

References

- Bai, M., Zhou, Z., Wang, R., Yang, Y., Qin, Z., Chen, Y., Mu, C., Liu, J., & Yu, D. (2023). HouYi: An open-source large language model specially designed for renewable energy and carbon neutrality field. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2308.01414>
- Bonner, E., Lege, R., & Frazier, E. (2023b). LARGE LANGUAGE MODEL-BASED ARTIFICIAL INTELLIGENCE IN THE LANGUAGE CLASSROOM: PRACTICAL IDEAS FOR TEACHING. *Teaching English With Technology*, 2023(1). <https://doi.org/10.56297/bkam1691/wieo1749>
- Cai, H., & Wu, S. (2023). TKG: Telecom Knowledge Governance Framework for LLM Application. *Research Square (Research Square)*. <https://doi.org/10.21203/rs.3.rs-3252192/v1>
- Chang, J., Brantley, K., Ramamurthy, R., Misra, D., & Sun, W. (2023b). Learning to generate better than your LLM. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2306.11816>
- Cheng, D., Huang, S., & Wei, F. (2023). Adapting large language models via reading comprehension. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2309.09530>
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., . . . Fiedel, N. (2022). PaLM: Scaling Language Modeling with Pathways. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2204.02311>
- Chowdhury, J. R., Zhuang, Y., & Wang, S. (2022). Novelty Controlled Paraphrase Generation with Retrieval Augmented Conditional Prompt Tuning. *Proceedings of the*

. . . *AAAI Conference on Artificial Intelligence*, 36(10), 10535–10544.

<https://doi.org/10.1609/aaai.v36i10.21297>

Darapaneni, N., Chetan, P., Paduri, A. R., Gaddala, A., Tiwari, G., Basu, S., & Parvathaneni,

S. (2021). Building a question and answer system for news domain. *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. <https://doi.org/10.1109/icsccc51823.2021.9478180>

De Curtò, J., De Zarzà, I., & Calafate, C. T. (2023). Semantic Scene Understanding with Large Language Models on Unmanned Aerial Vehicles. *Drones*, 7(2), 114.

<https://doi.org/10.3390/drones7020114>

Guo, Q., Cao, S., & Yi, Z. (2022). A medical question answering system using large language models and knowledge graphs. *International Journal of Intelligent Systems*, 37(11), 8548–8564. <https://doi.org/10.1002/int.22955>

Jeong, C. (2023). A study on the implementation of generative AI services using an Enterprise Data-Based LLM application architecture. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2309.01105>

Jiang, A. Q., Sablayrolles, A., Arthur, M., Bamford, C., Chaplot, D. S., De Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T. J., Lacroix, T., & Sayed, W. E. (2023). Mistral 7B. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2310.06825>

Krishna, C. S. M. (2023). Prompt Generate Train (PGT): few-shot domain adaption of retrieval augmented generation models for open book Question-Answering. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2307.05915>

Liu, Y., & Lapata, M. (2019). Text Summarization with Pretrained Encoders. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1908.08345>

- Manathunga, S., & Illangasekara, Y. A. (2023). Retrieval Augmented Generation and Representative Vector Summarization for large unstructured textual data in Medical Education. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2308.00479>
- Pavlyshenko, B. M. (2023). Financial News Analytics using Fine-Tuned Llama 2 GPT model. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2308.13032>
- See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.1704.04368>
- Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., & Nanayakkara, S. (2023). Improving the domain adaptation of Retrieval Augmented Generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics, 11*, 1–17. https://doi.org/10.1162/tacl_a_00530
- Su, J., Lu, Y., Pan, S., Wen, B., & Liu, Y. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2104.09864>
- Topsakal, O., & Akıncı, T. Ç. (2023). Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *Vol. 1 No. 1 (2023): International Conference on Applied Engineering And Natural Sciences, 1(1)*, 1050–1056. <https://doi.org/10.59287/icaens.1127>
- Touvron, H., Martin, L., Stone, K. H., Albert, P. J., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., . . . Scialom, T. (2023). Llama 2: Open foundation and Fine-Tuned chat models. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2307.09288>

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *arXiv (Cornell University)*, 30, 5998–6008. <https://doi.org/10.48550/arXiv.1706.03762>
- Wang, C., Ong, J., Wang, C. Y. J., Ong, H., Cheng, R. R., & Ong, D. (2023). Potential for GPT Technology to optimize future clinical Decision-Making using Retrieval-Augmented Generation. *Annals of Biomedical Engineering*.
<https://doi.org/10.1007/s10439-023-03327-6>
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., & Zettlemoyer, L. (2022). OPT: Open Pre-trained Transformer Language Models. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2205.01068>

Appendix A

System Testing

The system is visualized using a Web GUI and tested in the cases below. Each test will include a UI snapshot. The created UI is a simple interface with only a few functions, whereas Figure A1 checks the browse file button operation. Figure A1 depicts the functional testing of the File Upload activity. Figure A3 shows the Files uploaded on the left side of the UI. Figure A4 depicted the operation of the submit and process buttons.

Figure A1

Browse File Button Test

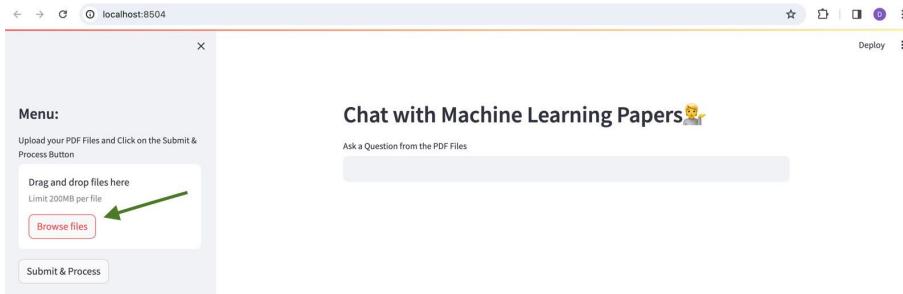


Figure A2

File Upload Test

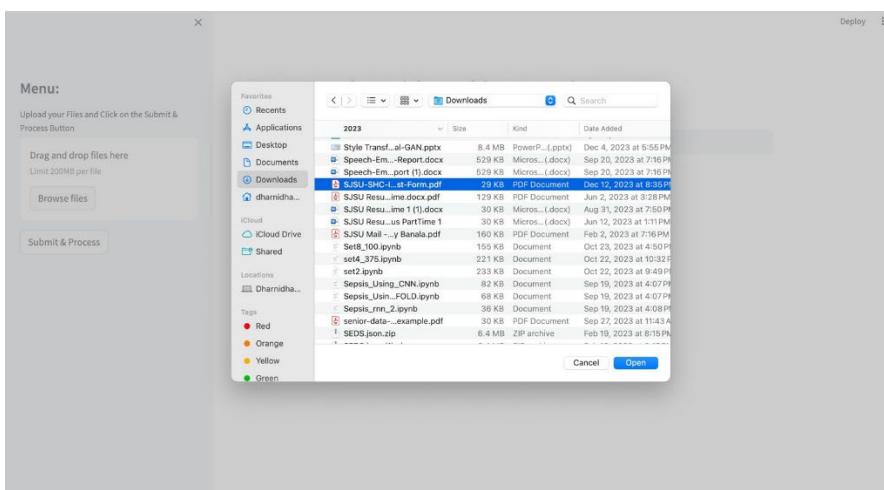
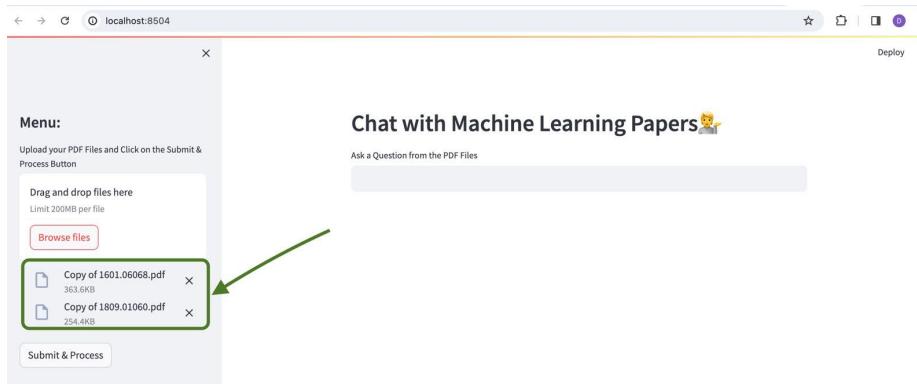
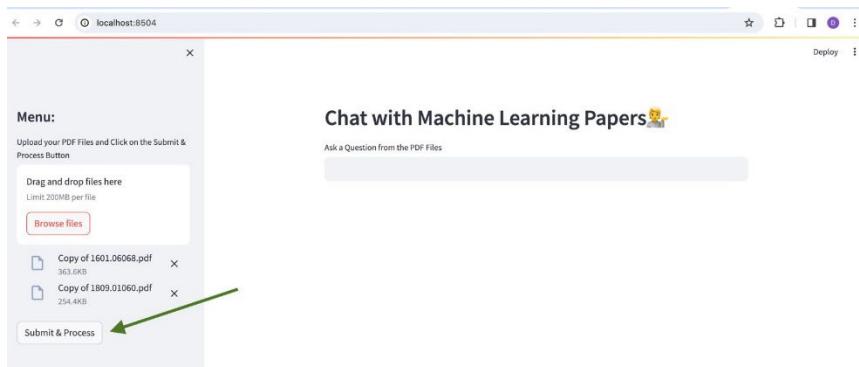


Figure A3

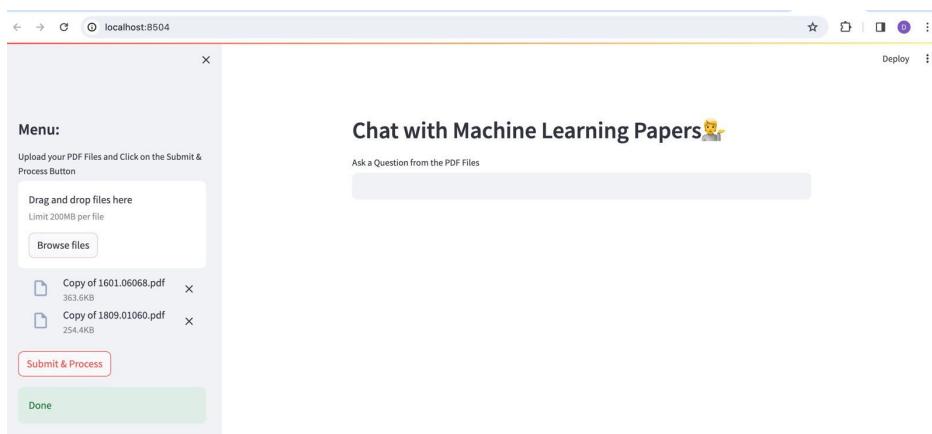
Are Files Uploaded? Test

**Figure A4**

Submit and Process Button Test

**Figure A5**

Submit and Process Button Working Test



Based on testing performed on 20 different file inputs with an average of 2-3 files at a time, the time required to perform data parsing is estimated to be up to 1.5 minutes when run on a GPU processor.

Appendix B

Project Data Source and Management

The process begins with Figure B1, which outlines an example of using Google Drive storage for ArXiv database papers. Moving forward, Figure B2 illustrates the creation of a script designed to extract Question and Answer sets from QASPER. Subsequently, Figure B3 showcases the final dataset after transformation from QASPER, indicating a significant step in data preparation.

Figure B4 displays the distribution of question lengths, providing insights into the variability within the dataset. Finally, Figure B5 demonstrates the utilization of fine-tuned models sourced from Hugging Face, highlighting the integration of advanced NLP techniques into the project.

Figure B1

PDF Files Stored and Managed in Google Drive

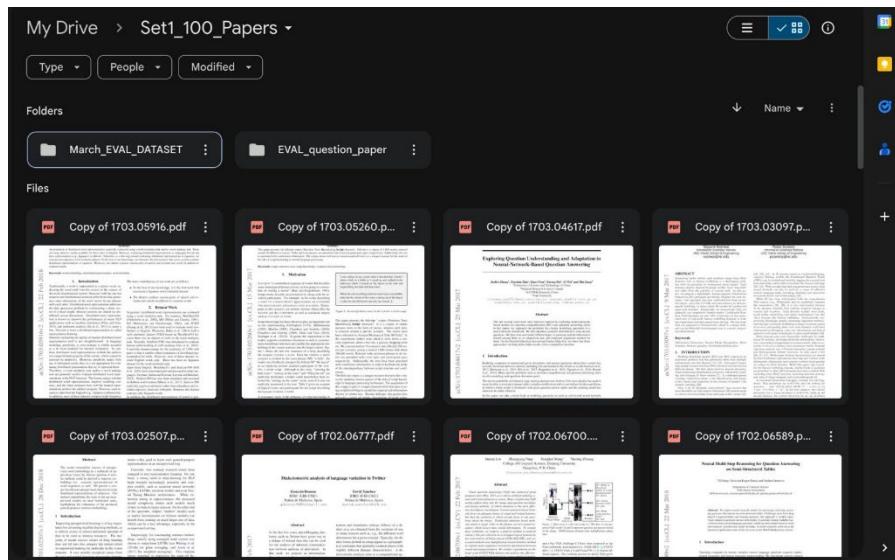


Figure B2

Script for Data Transformation of QASPER

```

1 import json
2 import pandas as pd
3 from tqdm import tqdm
4
5 def create_dataframe(dataset):
6     paper_ids, questions, yes_nos, answers, evidences, highlighted_evidences = [], [], [], [], [], []
7     for item in dataset:
8         paper_ids.append(item['paper_id'])
9         questions.append(item['question'])
10        yes_nos.append(item['yes_no'])
11        answers.append(item['answer'])
12        evidences.append('\n'.join(item['evidence']))
13        highlighted_evidences.append('\n'.join(item['highlighted_evidence']))
14
15    df = pd.DataFrame({
16        'paper ID': paper_ids,
17        'Question': questions,
18        'Yes/No Question': yes_nos,
19        'Answer': answers,
20        'Evidence': evidences,
21        'Highlighted Evidence': highlighted_evidences
22    })
23    return df
24

25 datasets = []
26 for mode in ['train', 'dev']: # Adjust these as per the available JSON files
27     with open(f'qasper-{mode}-v0.3.json', 'r') as file:
28         data = json.load(file)
29
30     dataset = []
31     for paper_id, paper_data in tqdm(data.items()):
32         for qa in paper_data['qas']:
33             question = qa['question']
34             for answer in qa['answers']:
35                 ans = answer['answer'] # Corrected this line to access the answer dictionary directly
36                 if not ans['unanswerable']:
37                     dataset.append({
38                         'paper_id': paper_id,
39                         'question': question,
40                         'yes_no': ans['yes_no'],
41                         'answer': ans['free_form_answer'] if ans['free_form_answer'] else "Yes" if ans['yes_no'] else "No",
42                         'evidence': ans['evidence'],
43                         'highlighted_evidence': ans['highlighted_evidence']
44                     })
45     datasets.extend(dataset)
46
47 # Remove duplicate questions
48 question_set = set()
49 unique_datasets = [item for item in datasets if item['question'] not in question_set and not question_set.add(item['question'])]
50
51 df = create_dataframe(unique_datasets)
52 print(df.head())
53
54 # Save to CSV
55 # df.to_csv('qa_dataset.csv', index=False)
56

```

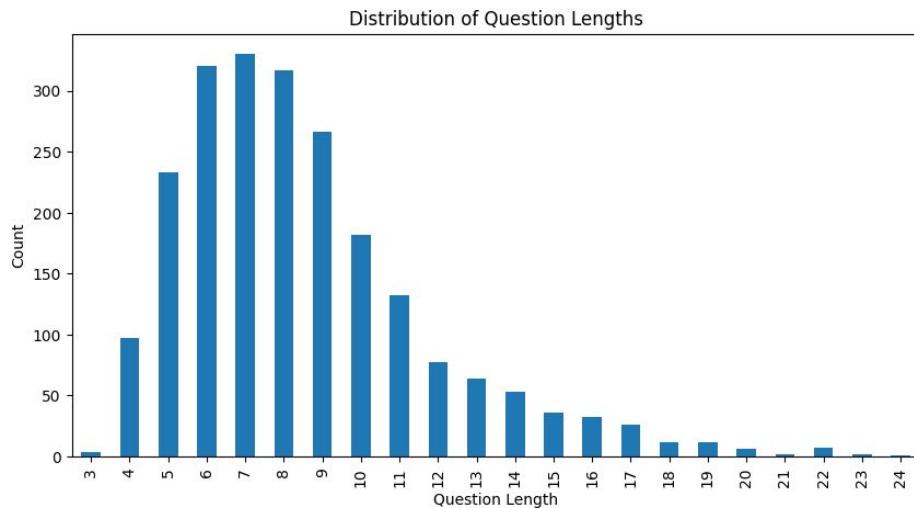
Figure B3*Final Transformed QASPER Dataset*

The screenshot shows a dataset viewer interface with the following details:

- Datasets:** kalyan003/QAS_Final_thor_version01
- Split:** train · 3.78k rows
- Text:** string · lengths
- 172-344** 32.28
- Content Preview:**
 - <>[INST] What is the seed lexicon? [/INST] The seed lexicon consists of positive and negative predicates. If the predicate of an extracted event is in the seed lexicon and does not involve complex phenomena like negation, we assign the corresponding polarity score (\$+1\$ for positive events and \$-1\$ for negative events) to the event. </>
 - <>[INST] What are the results? [/INST] FLOAT SELECTED: Table 3: Performance of various models on the ACP test set. </>
 - <>[INST] How are relations used to propagate polarity? [/INST] As illustrated in Figure FIGREF1, our key idea is that we can exploit discourse relations BIBREF4 to efficiently propagate polarity from seed predicates that directly report one's emotions (e.g., ,Äto be glad,Ä is positive). Suppose that events \$x_1\$ are \$x_2\$ are in the discourse relation of Cause (i.e., \$x_1\$ causes \$x_2\$).
 - <>[INST] How big is the Japanese data? [/INST] As a raw corpus, we used a Japanese web corpus that was compiled through the procedures proposed by BIBREF13. </>
 - <>[INST] What are labels available in dataset for supervision? [/INST] In this paper, we work on recognizing the polarity of an affective event that is represented by a score ranging from \$-1\$ (negative) to 1 (positive). </>
 - <>[INST] How big are improvements of supervised learning results trained on smaller labeled data enhanced with proposed approach compared to basic approach? [/INST] FLOAT SELECTED: Table 4: Results for small labeled training data. Given the performance with the full dataset, we show BERT trained only with the AL data. </>
 - <>[INST] How does their model learn using mostly raw data? [/INST] In this paper, we propose a simple and effective method for learning affective events that only requires a very small seed lexicon and a large raw corpus. As illustrated in Figure FIGREF1, our key idea is that we can exploit discourse relations BIBREF4 to efficiently propagate polarity from seed predicates that.
 - <>[INST] How big is seed lexicon used for training? [/INST] We constructed our seed lexicon consisting of 15 positive words and 15 negative words, as shown in Section SECREF27. </>
 - <>[INST] How large is raw corpus used for training? [/INST] As a raw corpus, we used a Japanese web corpus that was compiled through the procedures proposed by BIBREF13. </>
 - <>[INST] Does the paper report macro F1? [/INST] FLOAT SELECTED: Table 7: Recall and precision scores of the best model (dbmdz) for each emotion on the test set. ,ÄSupport,Ä signifies the number of labels. </>
 - <>[INST] How is the annotation experiment evaluated? [/INST] Figure FIGREF18 shows the confusion matrices of labels between annotators as heatmaps. </>
 - <>[INST] What are the aesthetic emotions formalized? [/INST] Deeply moved readers shed tears or get chills and goosebumps even in lab settings BIBREF4. In cases like these, the emotional response actually implies an aesthetic evaluation: narratives that have the capacity to move readers are evaluated as good and powerful texts for this very reason. Similarly, feelings of suspense.
 - <>[INST] Do they report results only on English data? [/INST] We compute our measures over the comments written by users in a community in time windows of months, for each sufficiently active month, and manually examine communities where the bulk of the contributions are in a certain language. </>
- Pagination:** < Previous 1 2 3 ... 38 Next >

Figure B4

Visualization of Questions Length

**Figure B5**

Huggingface Connection Test

The screenshot shows a Jupyter Notebook interface with a code cell containing Python code for connecting to a Huggingface model. The code imports `huggingface_hub` and `notebook_login`, logs in, and initializes an AutoTokenizer. A progress bar indicates the download of files from the URL `/usr/local/lib/python3.10/dist-packages/transformers/models/auto/tokenization_auto.py:757: FutureWarning`.

```
+ Code + Text
Step 10: Creating a Llama2 model wrapper
[ ] 1 from huggingface_hub import notebook_login
2 import torch
3
4 notebook_login()
5
Token is valid (permission: write).
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /root/.cache/huggingface/token
Login successful
[ ] 1 tokenizer = AutoTokenizer.from_pretrained("kalyan003/Llama-2-7b-chat-finetune-full-text",use_auth_t
/usr/local/lib/python3.10/dist-packages/transformers/models/auto/tokenization_auto.py:757: FutureWarning:
tokenizer_config.json: 100% [1.57k/1.57k, 00:00<00:00, 100kB/s]
tokenizer.json: 100% [1.84M/1.84M, 00:00<00:00, 5.60MB/s]
special_tokens_map.json: 100% [437/437, 00:00<00:00, 33.7kB/s]
```

Appendix C

Project Program Source Library, Presentation, and Demonstration

Table C1

Artifact Access with Description and Links

Artifact	Description	Link
Readme	Understanding various steps to go through the code and implement	https://docs.google.com/document/d/1Sjv6nfYm8XP0rzZNWxbNYp4rJaBgcwS/edit?usp=drive_link&ouid=107928492007816352469&rtpof=true&sd=true
Fine-tuned Llama 2	Custom Fine-tuned Llama model artifacts	https://huggingface.co/kalyan003/Llama-2-7b-chat-finetune-full-text
Fine-tuned Mistral 7B	Custom Fine-tuned Mistral 7B model artifacts	https://huggingface.co/kalyan003/Mistral-2-7b-chat-finetune-QEHE/tree/main
Arxiv Papers	Datasets from Arxiv Papers	https://drive.google.com/drive/folders/1l-BUIV45K1gTBljBzSA4K9tPlk46ejnl?usp=drive_link https://docs.google.com/spreadsheets/d/1cDesg6cWM2RX5MuvrLEOLdIJ_zD-3-hS/edit?usp=drive_link&ouid=107928492007816352469&rtpof=true&sd=true
Transformed QASPER	QASPER dataset used for Fine-tuning	https://github.com/Dharnidhar72/Comprehending-ML-Papers-using-Llama7B https://drive.google.com/drive/folders/1OVPv037Zqt2igTlba0hzzHyXI_Jbz2yG?usp=drive_link
Source Code	Github Link For Source Code	https://github.com/Dharnidhar72/Comprehending-ML-Papers-using-Llama7B
298B Presentations	Submitted presentations	https://drive.google.com/drive/folders/1ID7Bez1YhgkG16_YhcnrwH7I2QS1QzVJ?usp=drive_link
Demo	Recordings of a working interface of the system	https://drive.google.com/drive/folders/1ID7Bez1YhgkG16_YhcnrwH7I2QS1QzVJ?usp=drive_link