I in specific areas for further optimization.

Applied Data Science Department

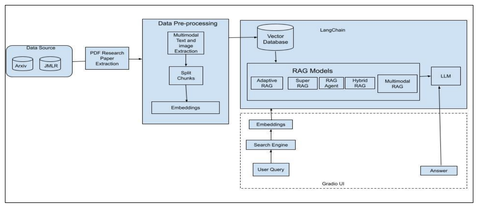
Large-Scale PDF Analysis Leveraging RAG and LLMs for Enhanced ML Literature Insights

Project Advisor: Simon Shim

Badrinarayanan, Shruti
Neelamegam, Divya
Suresh Kumar, Sourabh
Venkatram, Poojitha

Introduction

The field of Machine Learning and AI is rapidly advancing, making it challenging for researchers, students, and professionals to stay updated with the latest developments. This application simplifies the accessibility and comprehension of ML and AI literature, addressing the need for efficient knowledge transfer and collaboration within the ML community for students and researchers.



Project Objectives

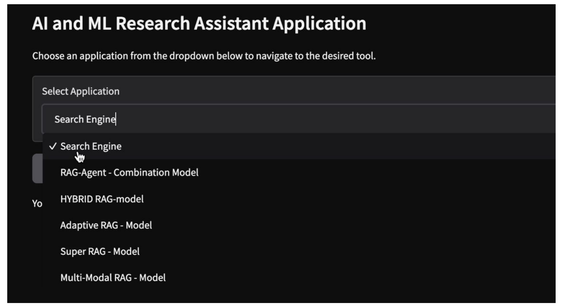
1. Develop a chat application using RAG and LLMs to interpret research paper PDFs
2. Parse non-text content in PDFs (multimodal content)
3. Provide informed responses to technical questions using a database of 25000+ research papers from arXiv and JMLR.
4. Create an ML/AI Research Paper Search Engine
5. Identify optimal RAG components for effective performance

Methodology

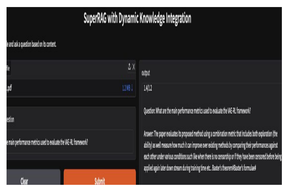
1. **Search Engine**: Efficiently retrieves relevant research papers from the indexed datasets, serving as a robust tool for academic and professional queries and a baseline for advanced retrieval models.
2. **RAG-Agent (Combination Model):** Combines retrieval-augmented generation with advanced routing mechanisms to Adaptive/Corrective/Fallback RAGs, leveraging LLaMa-3 for superior accuracy, contextual relevance, and balanced resource usage.
3. **Hybrid RAG:** Integrates structured and unstructured data retrieval, ensuring strong relevance and groundedness while optimizing GPU and CPU resource usage.
4. **Adaptive RAG:** Dynamically routes queries to relevant modules, prioritizing context relevance and answer correctness for flexible query handling along with a fine-tuned Mistral LLM.
5. **Super RAG:** Excels in context and answer relevance, offering high responsiveness for applications prioritizing precision, though slightly limited in groundedness.
6. **Multi-Modal RAG:** Handles diverse data types like text and images, enabling balanced performance across metrics for multimodal applications, with higher CPU usage.

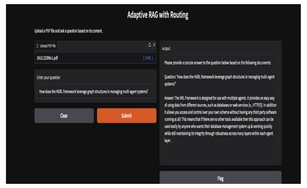
Analysis and Results

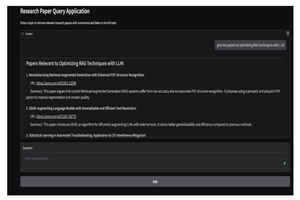
The application serves as a centralized platform for researchers, offering multiple advanced tools for AI and machine learning tasks. The dropdown menu allows users to navigate effortlessly between various models and functionalities, such as a dedicated Search Engine, Hybrid RAG models, Adaptive RAG, Super RAG, Multi-Modal RAG, and a combination model powered by RAG-Agent. These tools streamline the research process, supporting tasks ranging from data retrieval to multimodal analysis, enhancing efficiency and precision in AI-driven workflows.

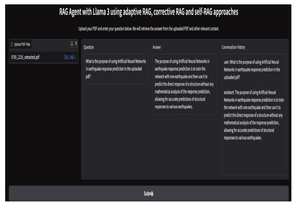


The image below highlights four interfaces showcasing advanced Retrieval-Augmented Generation (RAG) applications. These include tools for dynamic PDF querying, adaptive routing for contextual responses, efficient research paper summarization, and a versatile RAG agent powered by Llama 3 with adaptive and corrective methods. Each interface emphasizes precise, context-aware knowledge retrieval to enhance user experience and streamline access to complex information.



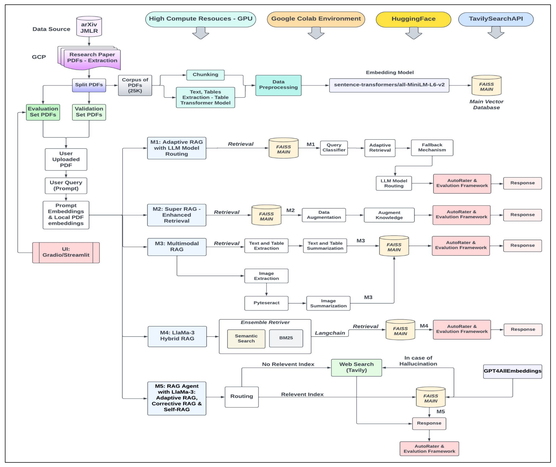






The architecture is a sophisticated Retrieval-Augmented Generation (RAG) framework leveraging diverse data sources like research paper PDFs, processed through advanced embedding models (e.g., sentence-transformers) and stored in a FAISS-based vector database. It incorporates multiple RAG approaches—Adaptive, Super, Multimodal, and Hybrid—optimized with routing, augmentation, and ensemble retrieval techniques.

Integration with evaluation frameworks ensures robust responses, while fallback mechanisms like Tavily Search handle queries outside of local and main databases, ensuring accurate and contextually relevant outputs. The framework supports diverse user queries via a streamlined UI, utilizing cutting-edge AI tools such as LLaMa-3, PyTesseract, and GPT-4All embeddings

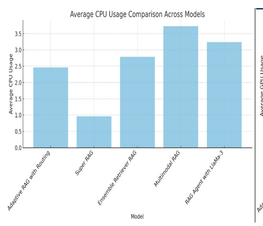


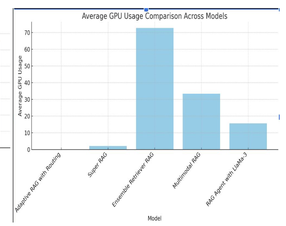
The CPU usage comparison shows that Multimodal RAG and RAG Agent with LLaMa-3 have the highest CPU consumption, while Super RAG is the most efficient in terms of CPU usage.

Comparison of Overall Resource Utilisation Metrics *(Original from Team 8)*

Model	Response Time	CPU start usages	CPU end usages	Average CPU usages	Average GPU usages
Adaptive RAG with Routing	60.62	2.125	2.8	2.462	0
Super RAG	5.298	3.106	0.9625	2.034	75.564
Ensemble Retriever RAG	10.16	3.20	3.156	2.786	72.72
Multimodal RAG	5.43	4.47	4.82	3.73	33.36
RAG Agent with LlaMa-3	6.86	3.08	3.067	3.237	15.63

The GPU usage comparison reveals that Ensemble Retriever RAG has the highest GPU consumption, while RAG Agent with LLaMa-3 is the most efficient. In contrast, the CPU usage chart shows Multimodal RAG and RAG Agent with LLaMa-3 utilizing higher CPU resources, with Super RAG being the least demanding





RAG Correctness Metrics

Model	Context relevance	Answer Relevance	Groundedness	Answer Correctness	Human Judge Score
Adaptive RAG with Routing	9.0	9.0	7.0	9.017	7.437
Super RAG	9.0	8.625	6.0	7.0	7.75
Ensemble Retriever RAG	6.08	7.5	8.16	7.27	7.72
Multimodal RAG	7.1	7.4	7.30	6.90	7.11
RAG Agent with LlaMa-3	8.87	8.95	8.755	8.41	8.77

Summary/Conclusions

The RAG Agent with LLaMa-3 emerged as the top-performing model, achieving the highest scores across all metrics, including an impressive 8.77 human judge score. Adaptive RAG with Routing demonstrated exceptional performance in context relevance and answer correctness, while Super RAG and Ensemble Retriever RAG excelled in answer relevance and groundedness, respectively. Multimodal RAG leverages multiple data modalities, showing promise for broader applications and future improvements.

Key References

[1] Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J. C. (2024). Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.
[2] Yan, S. Q., Gu, J. C., Zhu, Y., & Ling, Z. H. (2024). Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.
[3] Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.
[4] Thakur, A., & Gupta, R. (2024). Introducing Super RAGs in Mistral 8x7B-v1. *arXiv preprint arXiv:2404.08940*.
[5] Hsieh, C. Y., Li, C. L., Yeh, C. K., Nakhost, H., Fujii, Y., Ratner, A., ... & Pfister, T. (2023). Distilling step-by-step! outperforming large language models with less training data and smaller model sizes. arXiv preprint arXiv:2305.02301. https://arxiv.org/abs/2305.02301

Acknowledgements

We would like to express our sincere gratitude to Dr. Simon Shim for his invaluable guidance and support throughout this project. We also extend our appreciation to our team members for their collaboration and dedication in bringing this project to fruition. Lastly, we acknowledge the contributions of the research community and the resources provided by databases such as Arxiv and the Journal of Machine Learning Research, which were essential for our work.