MILWAUKEE SCHOOL OF ENGINEERING
COMPUTER SCIENCE AND SOFTWARE ENGINEERING DEPARTMENT

INDEPENDENT STUDY REQUEST

Date: 08/29/24 Adam Haile requests to enroll in:
(Student Name)

Undergraduate

☑ CSC-4999 Computer Science Independent Study for 3 credits

☐ SWE-4999 Software Engineering Independent Study for _____ credits

Graduate

☐ CSC-6999 Computer Science Independent Study for _____ credits

Project Title: Developing Scalable RAG Systems in High Performance Cc

Project faculty advisor will be: Dr. Josiah Yoder

Registration will be for the Fall semester.
Note: ALL work must be completed during this semester.

Attach to this form:

- Student learning outcomes
- Description of the project or course of study
- Proposed method of solution
- Deliverables, with due dates
- Grading criteria (as discussed with faculty advisor)

NOTE: Form must be completed 6 weeks prior to the start of the semester of enrollment and presented at time of registration.

Approved by:

Josiah Yoder — *Faculty Advisor* — 8/29/24 *Date*

Derek Riley Digitally signed by Derek Riley Date: 2024.08.29 15:03:16 -05'00' — *Program Director* — 8/29/2024 *Date*

CSSE *Department Chair* — 2024-08-30 *Date*

PROVIDE COPIES TO:

*Registrar's Office
*Student
*Faculty Advisor
*Program Director
*CSSE Office - csserequest@msoe.edu

Note that independent studies scheduled for summer terms are approved only in extraordinary circumstances.

Milwaukee School of Engineering
CSC-4999 Independent Study
Developing Scalable RAG Systems in High Performance Computing
Adam Haile
August 28, 2024

Learning Outcomes

- Ability to analyze and improve on existing Retrieval-Augmented Generation (RAG) implementations within high-performance computer environments.
- Skills in building and optimizing efficient RAG pipelines, including vector databases and retrieval systems.
- Expertise in designing scalable, containerized architectures for flexible deployment in high-performance environments.
- Proficiency in creating clear, user-friendly documentation for deploying and managing complex computing systems.

Project/Course of Study

Investigation and Analysis of RAG Solutions

- Conduct a detailed analysis of the architecture and methodologies used in prior RAG solutions hosted on ROSIE, with a focus on the 2023-2024 ROSIE competition-winning RAG system.
- Document insights gained from their design and implementation process, highlight areas of focus on new developments needed.

Design of a Scalable RAG Architecture

- Develop a detailed design for a new, scalable RAG architecture tailored to the needs of the ROSIE environment. This design will include components for document ingestion, chunking, embedding, vector database management, and retrieval.
- Plan a containerization approach that ensures the RAG service can be deployed and managed efficiently within ROSIE.

Implementation of High-Performance Systems for RAG Applications

- Develop and implement the proposed vector database solution optimized for integration with ROSIE, leveraging existing technologies and best practices.

- Build a fully functional RAG pipeline, including embedding models, document processing, and retrieval mechanisms. The pipeline will be optimized for both speed and accuracy.
- Containerize the RAG system for deployment on ROSIE, ensuring seamless operation, scalability, and user accessibility.

Proposed Method of Solution

The initial investigation will be an investigation of the 2023-24 ROSIE competition winners RAG solution, analyzing areas where code-reuse can be used and segments where new development will be required to conform with the on-demand architecture.

Additional sources for implemented vector databases and RAG architecture will primarily consist of online articles from proprietary vector database providers and research papers.

- Lewis, P., et al. (2021, April 12). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. ArXiv.org. https://doi.org/10.48550/arXiv.2005.11401
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023, December 18). Retrieval-Augmented Generation for Large Language Models: A Survey. ArXiv.org. https://doi.org/10.48550/arXiv.2312.10997
- Building a complete RAG pipeline and system: https://huggingface.co/learn/cookbook/en/rag_with_hugging_face_gemma_mongodb
- Brief introduction into Vector Databases: https://www.nvidia.com/en-us/glossary/vector-database/
- Implementing a pgvector database: https://www.timescale.com/blog/postgresql-as-a-vector-database-create-store-and-query-openai-embeddings-with-pgvector/
- Knowledge Base Chatbot Workshop (from Data Driven WI): https://workshop.tyler-faulkner.com/

Deliverables

Week 3

- Technical Report 1 – An analysis of the pre-existing RAG solutions and their techniques/methodologies, highlighting areas where current solutions will need to be improved to meet the demands of the final system.

Week 7

- Technical Report 2 – Detailed proposal of a new RAG architecture, including management of the complete pipeline for ingestion, chunking, embedding, vector database management, and retrieval, all within a Docker container.

Week 12

- Live demonstration of the functional RAG system running within ROSIE, showcasing core features and capabilities. This demonstration will include performance benchmarks and a comparison with previous solutions to highlight improvements.

Final

- Comprehensive final report detailing the final architecture, development challenges. Additionally, the complete documentation of a high-quality user guide for utilizing the RAG system within ROSIE.

Grading Criteria

- Technical Report 1 – 20%
- Technical Report 2 – 20%
- Demonstration – 20%
- Final implementation and report – 30%
- Student initiative and professionalism – 10%