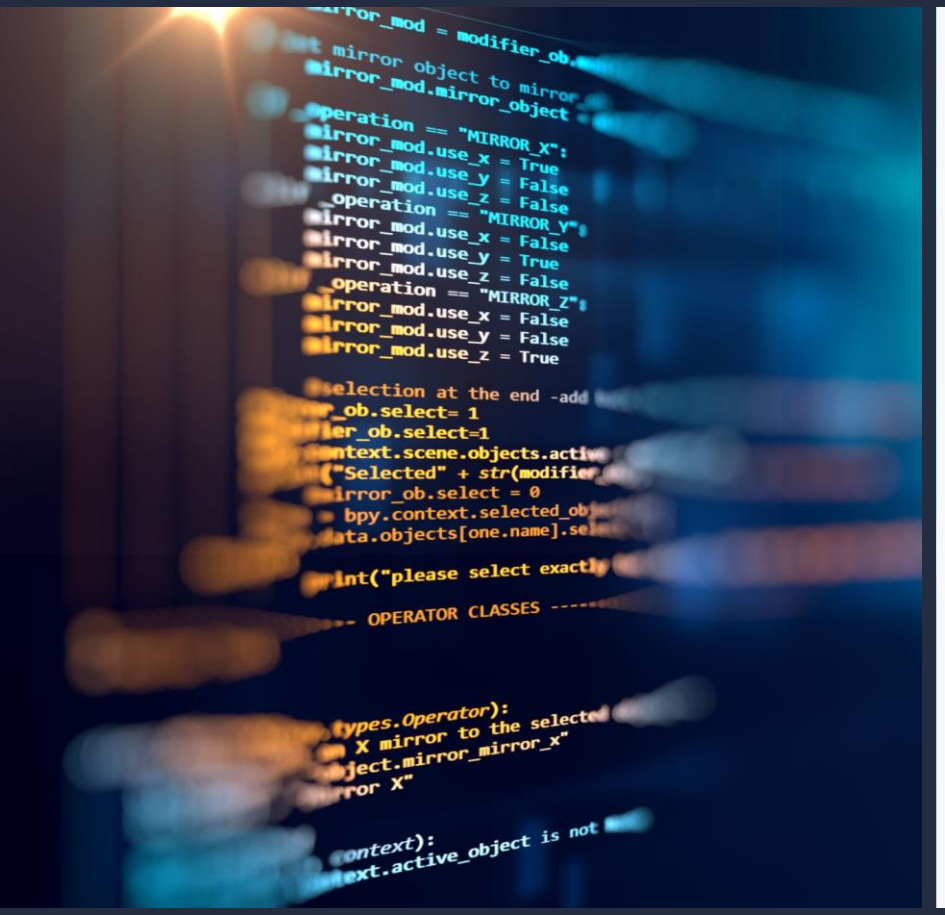

HACKEREARTH RAG APPLICATION – SPRINT 3

Ethan Villalovoz, Molly Iverson, Adam Shtrikman, Chandler Juego

Introduction

Develop a RAG (Retrieval-Augmented Generation) application for HackerEarth that will utilize vector search, knowledge graphs, and a LLM to answer questions and generate content from a knowledge base of more than 10,000 Wikipedia articles.

Sprint Objectives

Pipeline Integration (Main Objective): Establish a fully functional query-processing pipeline that integrates vector search (VS), large language model (LLM), natural language processing (NLP), and knowledge graph (KG) components.

Vector Search and LLM: Develop a system that retrieves contextualized results using vector search and processes them through an LLM.

Integration of Components: Combine outputs from the knowledge graph and vector search with the LLM in the query pipeline.

NLP and KG Integration: Automate SPARQL query generation from NLP output and connect the NLP and KG modules to the system interface.

CI/CD Pipeline: Implement a CI/CD pipeline to automate testing development workflows.

Feature Implementation

Team Member	Feature	Objective
Ethan	- Vector Search Component - LLM Component	- To get contextualized queries with VS results - Run queries through an LLM and receive a clear response
Chandler	-Vector Search and LLM Integration	- Combine KG and VS outputs to form a contextualized query - Run the frontend with the LLM integrated
Molly	- NLP and KG Integration	- Take NLP output to create a SPARQL query for KG - Connect the NLP and KG pipeline to the frontend
Adam	- CI/CD Pipeline for Testing Automation	- Automate development testing

Feature Demos

Testing and Validation

Integration Tests

- NLP tasks
  - Tokenization
  - Entity extraction from a query (for KG querying)
  - Harmful intent check

Unit Tests

- Knowledge graph
  - DBPedia querying
- Vector search
  - Loading embeddings
  - Building vector index
  - Search

Kanban Overview & Contributions

- **Molly**
  - In-Class Presentation
  - Integration of NLP and KG pipeline
  - Future work report section
- **Ethan**
  - Client Presentation
  - Vector Search and LLM code
  - Alpha prototype description report section
- **Chandler**
  - In-Class Presentation
  - Integration of VS and LLM pipeline
  - Alpha prototype demo report section
- **Adam**
  - Client Presentation
  - CI/CD pipeline for testing automation
  - Alpha prototype description report section
  - Sprint report
- **All**
  - Meeting notes for client meetings

Capstone Agile Planning

Backlog | Priority board | Team items | Roadmap | In review | My items | + New view

Filter by keyword or by fie

Backlog 0 / 20 Estima

This item hasn't been started

Integrate VS and LLM into RAG pipeline and connect to React #67

Edit

Closed | #74 | mollyiverson/ACME10-HE-RAGApp Private

mollyiverson opened 3 weeks ago · edited by chandyego84 — Edits

As a developer or client
I need the React UI to handle user inputs effectively, send them to the NLP or directly to VS as appropriate, and return combined KG (Knowledge Graph) and VS results to an LLM (Large Language Model) for response generation
So that users can seamlessly interact with the system and get accurate, insightful responses in the chat interface

Details and Assumptions

- The React UI can send the user input to NLP and then VS or straight to VS
- Please send both KG and VS results to LLM, then display the results from there to the chat

Acceptance Criteria

```
Given the React UI can send the user input to NLP and then VS or directly to VS
When both KG and VS results are sent to the LLM for processing.
Then  chat interface will display the results returned from the LLM.
```

mollyiverson added enhancement 3 weeks ago

mollyiverson added this to Capstone Agile Planning 3 weeks ago

mollyiverson moved this to Ready in Capstone Agile Planning 3 weeks ago

mollyiverson assigned chandyego84 3 weeks ago

chandyego84 mentioned this 5 days ago
Vector Search Improvements, Vector Search + LLM Pipeline, File/Folder Refactoring #74

mollyiverson moved this from In progress to In review in Capstone Agile Planning 4 days ago

chandyego84 closed this as completed in #74 4 days ago

Assignees: chandyego84

Labels: enhancement

Projects: Capstone Agile Planning
Status: Done
Priority: Choose an option
Size: Choose an option
Estimate: 7
Start date: No date
End date: No date
Sprint: Sprint 3 • Nov 2 - Nov 28

Milestone: No milestone

Development: Vector Search Improvements, Vector Search + LLM Pipeline, File/Folder Refactoring
mollyiverson/ACME10-HE-RAGApp

Notifications — Customize
Unsubscribe
You're receiving notifications because you're subscribed to this thread.

Estimate: 0

Done 65 / 75 Estimate: 152

This has been completed

ACME10-HE-RAGApp #71
Write alpha prototype description
3 | Sprint 3

ACME10-HE-RAGApp #68
Add team bio and project role information
1 | Sprint 3

ACME10-HE-RAGApp #69
Write Future Work section in project report
3 | Sprint 3

ACME10-HE-RAGApp #63
Implement LLM component
5 | Sprint 3

ACME10-HE-RAGApp #50
Generate text embeddings for Wikipedia dataset
4 | Sprint 2

ACME10-HE-RAGApp #62
Implement vector search with small subset of Wiki data
3 | Sprint 3

ACME10-HE-RAGApp #65
Set up CI/CD Pipeline for Testing Automation

+ Add item

Documentation of Prototype Report

Sprint Achievements & Challenges

Achievements

- **Alpha Prototype Description Completed:**
  - Finalized and condensed the description within the report's page constraints.
  - Moved excessive details to appendices to maintain clarity and focus.
- **Implemented LLM Handler:**
  - Leveraged LLaMA for generating natural language responses.
  - Successfully combined Knowledge Graph (KG) and Vector Search (VS) outputs.
- **Vector Search Functionality Integrated:**
  - Used FAISS to optimize search results based on Wikipedia embeddings.
- **CI/CD Pipeline Setup:**
  - Automated dependency installation and testing via GitHub Actions.
  - Improved testing consistency across the development process.
- **Backend and Frontend Integration:**
  - Linked VS, LLM, and KG components to the React UI.

Sprint Achievements & Challenges

Challenges and Solutions

- **Challenge: Integration of VS and LLM into the RAG pipeline.**
  - Solution: Modularized handlers and streamlined the data flow to ensure compatibility.
- **Challenge: Optimizing query responses from the LLM for context relevance.**
- Solution: Implemented cosine similarity and debugging messages for better accuracy.

Sprint Achievements & Challenges

Progress from Sprint 2

- Enhanced vector search with cosine similarity for improved accuracy.
- Established an LLM endpoint and integrated KG, VS, and React UI for a cohesive pipeline.

Sprint Retro

What Went Well:

- Initial prototype demonstrated core RAG functionality
- Robust testing: unit, integration, and automated testing with CI/CD pipeline

Improvements Needed:

- See Client Feedback

Client Feedback

Feedback:

- Focus on RAG pipeline, not full-stack features
- Ensure scalability for 10,000+ embeddings
- Enhance response speed and context relevance

Actions Taken:

- Removed plans for user authentication and chat session storage
- Adjusted next semester’s plans

Next Steps

Upcoming Tasks:

- Generate and Store 10,000 embeddings
- Implement chunking for better data embedding and retrieval
- Test the system with custom input
- System testing
- Explore different approaches to the retrieval part of RAG
- Improve NLP

Conclusion

Key Progress:

- Alpha prototype successfully demonstrated core capabilities
- Plans adjusted to prioritized system optimization and new retrieval approaches

Future Direction:

- Embed full Wikipedia dataset
- Optimize RAG pipeline performance

Thank You!

Please let us know if you have more questions or suggestions