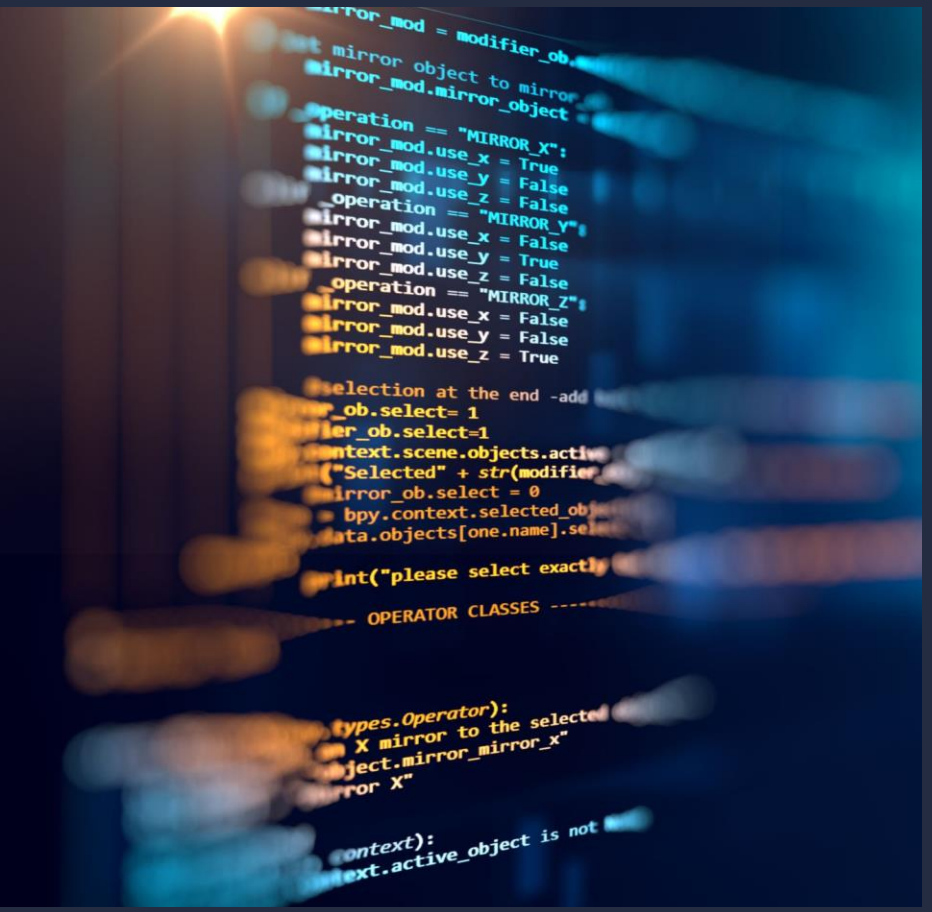


HACKEREARTH RAG APPLICATION – SPRINT 4

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

Performance Optimization: Collect performance data to address any bottlenecks, enhance response speed

Improved Data Readiness: Process embeddings and chunk dataset to improve RAG model accuracy and quality

Custom Use-Cases: Begin inputting custom datasets to test the robustness of our model

Documentation and Reporting: Continue realigning priorities and goals with client

Feature Implementation

Team Member	Feature	Impact
Ethan	Optimized LLM Performance	Switched from LLaMA 2.7B to ChatGPT API for faster and more reliable responses
Chandler	Implemented Document Chunking w/Vector Search	Improved vector search accuracy and efficiency by refining how documents are split into smaller, searchable units
Molly	System Testing & Performance Analysis	Conducted timing analysis on each component, identified bottlenecks, and documented performance improvements
Adam	Integrated Custom Dataset	Added class notes as an additional dataset, enhancing the ability to personalize embeddings and refine search results

Results

Initial Test

Component	Call 1 (ms)	Call 2 (ms)	Call 3 (ms)	Average (ms)
NLP Handler	552.30	346.40	552.00	483.57
Knowledge Graph	1137.30	1037.40	1231.70	1135.17
Vector Search	4418.30	7159.30	2973.20	4850.27
LLM	Not working	Not working	Not working	Not working
Total	6107.9	8543.1	4756.9	6469.3

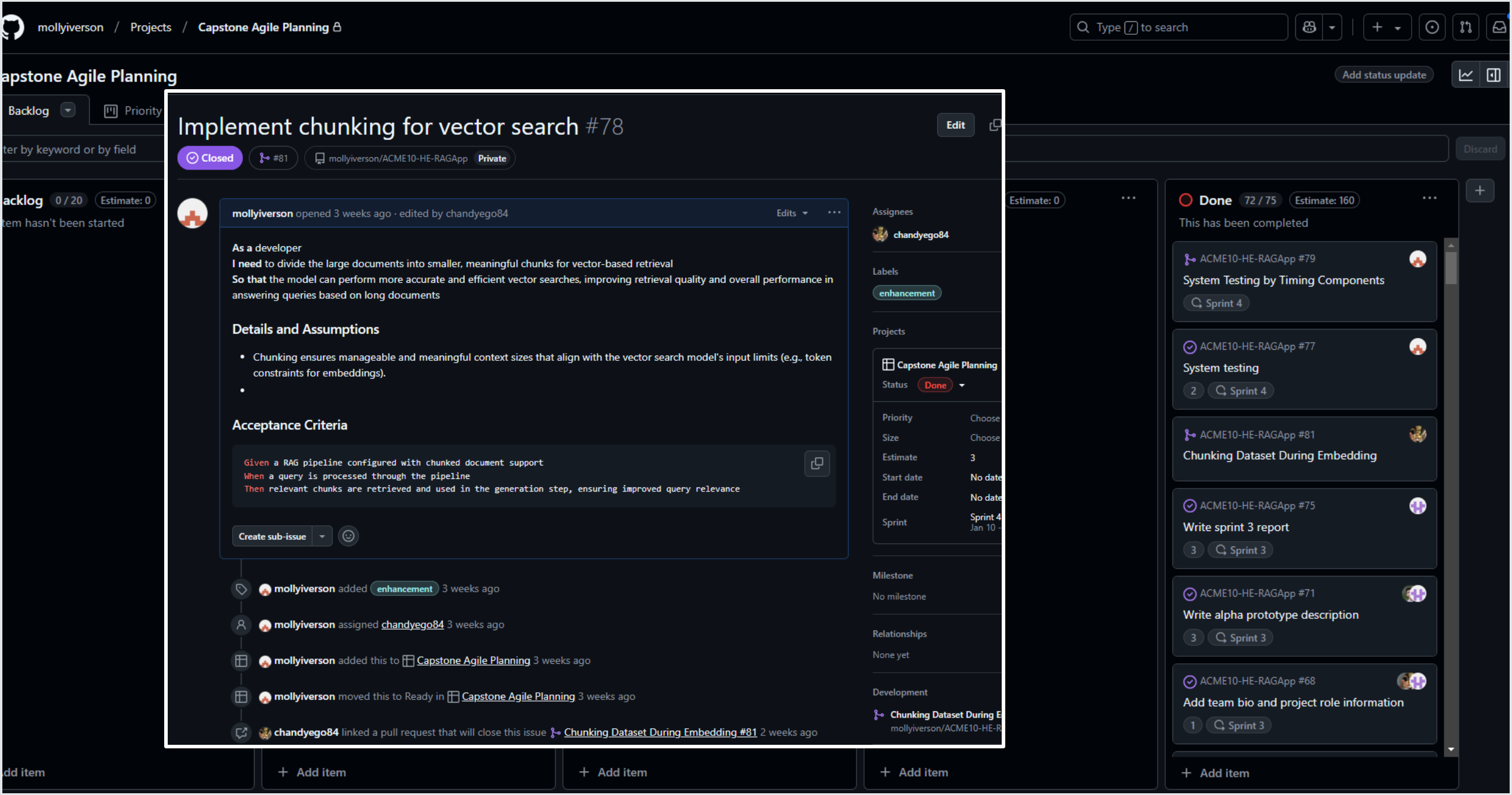
Change 1: Switching to ChatGPT and Chunking with Embeddings

Component	Call 1 (ms)	Call 2 (ms)	Call 3 (ms)	Average (ms)
NLP Handler	341.40	34.80	345.30	240.50
Knowledge Graph	1149.70	931.70	2112.30	1397.90
Vector Search	1924.70	1747.40	1753.50	1808.53
LLM	2405.30	2866.90	3245.50	2839.23
Total	5821.90	5581.30	7457.50	6286.90

Feature Demos

Kanban Overview & Contributions

- **Molly**
  - Conducted system testing by timing each component and documented performance and bottlenecks
  - Wrote the new project report
- **Ethan**
  - Optimized LLM performance by switching from LLaMA 2.7b to ChatGPT
  - Generated embeddings for the entire dataset
  - Sprint 4 report
- **Chandler**
  - Implemented document chunking for improved vector search accuracy and efficiency
- **Adam**
  - Integrated custom dataset (class notes) into embeddings workflow for personalized embedding generation
- **All**
  - Meeting notes for client meetings


mollyiverson / Projects / Capstone Agile Planning
Type / to search
apstone Agile Planning
Add status update
Backlog
Priority
ter by keyword or by field
Discard
acklog 0 / 20 Estimate: 0
tem hasn't been started
Estimate: 0
Done 72 / 75 Estimate: 160
This has been completed
Implement chunking for vector search #78
Edit
Closed
#81
mollyiverson/ACME10-HE-RAGApp Private
mollyiverson opened 3 weeks ago · edited by chandyego84
Edits
As a developer
I need to divide the large documents into smaller, meaningful chunks for vector-based retrieval
So that the model can perform more accurate and efficient vector searches, improving retrieval quality and overall performance in answering queries based on long documents
Details and Assumptions
Chunking ensures manageable and meaningful context sizes that align with the vector search model's input limits (e.g., token constraints for embeddings).
Acceptance Criteria
Given a RAG pipeline configured with chunked document support
When a query is processed through the pipeline
Then relevant chunks are retrieved and used in the generation step, ensuring improved query relevance
Create sub-issue
mollyiverson added enhancement 3 weeks ago
mollyiverson assigned chandyego84 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
chandyego84 linked a pull request that will close this issue Chunking Dataset During Embedding #81 2 weeks ago
Assignees
chandyego84
Labels
enhancement
Projects
Capstone Agile Planning
Status Done
Priority Choose
Size Choose
Estimate 3
Start date No date
End date No date
Sprint Sprint 4 Jan 10 -
Milestone
No milestone
Relationships
None yet
Development
Chunking Dataset During E
mollyiverson/ACME10-HE-R
ACME10-HE-RAGApp #79
System Testing by Timing Components
Sprint 4
ACME10-HE-RAGApp #77
System testing
2 Sprint 4
ACME10-HE-RAGApp #81
Chunking Dataset During Embedding
ACME10-HE-RAGApp #75
Write sprint 3 report
3 Sprint 3
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
dd item
Add item
Add item
Add item
Add item

Project Report Refinements

Evidence of Client meetings

Three meetings in Sprint 4:

- Jan 17th
- Jan 24th
- Feb 7th

Evidence of Client meetings

1/17/2025

2. Meeting Summary

Introduction:

- Started with informal chatting and catching up.
- Reviewed the capstone project goals, emphasizing interests and achievable milestones.

Client's Requirements:

- The client emphasized encouraging team interests and recognized that a full-fledged application may be difficult to achieve within the timeline. Focus should remain on delivering key components.
- With only two coding sprints remaining, it was suggested to focus on core deliverables and structure tasks accordingly.

Key Discussion Points:

- **Task Prioritization**: Focus on the first three tasks from the last semester task board, spreading them across the remaining sprints. Other tasks can be deprioritized if time is limited.
- **Wikipedia Embeddings**: Current progress on embedding the full Wikipedia text was reviewed. Suggested increasing the dataset size beyond the current 100 rows to test performance improvements.
- **Vector Database Storage**: Discussed pipeline clarifications for vector database storage and confirmed that original queries will be used for the LLM. Addressed slight confusion on whether embeddings (numerical data) were being passed directly to the LLM.
- **Final Deployment Goals**: Aim for project results to include some form of deployment.

Decisions Made:

- Increase the dataset size for embeddings beyond 100 rows and evaluate performance.
- Focus on completing the first three tasks of the last semester task board in the remaining sprints.
- Deprioritize lower-priority tasks unless additional time allows.
- Clarify pipeline functionality for vector database storage and ensure proper query handling.

Action Items:

- **Action 1**: Complete the current sprint – Due by 2/10/2025.
- **Action 2**: Add full embeddings to the README and test new performance – Due by 2/10/2025

Evidence of Client meetings

1/24/2025

2. Meeting Summary

Introduction:

- Started with informal chatting and catching up.
- Reviewed the goals for this sprint and overall project direction.

Client's Requirements:

- The client emphasized focusing on core functionality rather than a fully polished application. The primary objective is to develop a solid foundation for retrieval-augmented generation (RAG) with key components working efficiently.
- Given that only two coding sprints remain, the client advised structuring tasks strategically to maximize progress.

Key Discussion Points:

- **Sprint Priorities:** Reviewed the main deliverables for this sprint, ensuring they align with long-term project goals.
- **Team Member Contributions:**
  - **Adam:** Expanding dataset creation beyond Wikipedia documents to provide more diverse and useful retrieval data.
  - **Chandler:** Researching and implementing optimal chunking strategies for processing large datasets efficiently.
  - **Molly:** Setting up and refining system testing to validate different components of the application.
  - **Ethan:** Transitioning from LLaMA to ChatGPT for improved response generation and integration with existing architecture.
- **Scaling Wikipedia Embeddings:** Discussed increasing the number of rows used in embeddings beyond the initial 100-row limit to evaluate improvements in performance and accuracy.
- **System Pipeline & Storage:** Clarified the role of vector database storage within the RAG pipeline, ensuring the pipeline retains original queries for the LLM while effectively utilizing embeddings for retrieval.

Decisions Made:

- **Expand Dataset Scope:** Increase dataset size for embeddings and analyze performance improvements.
- **Maintain Strategic Focus:** Prioritize first three tasks from the last semester's task board for this sprint while keeping lower-priority tasks flexible.
- **Finalize LLM Transition:** Replace LLaMA with ChatGPT for response generation.
- **Refine Testing and Debugging:** Continue strengthening system testing for robustness.

Sprint Achievements & Challenges

Achievements:

- **Faster, higher-quality responses**: Switched to more optimal LLM
- **System Testing**: Began crucial step of performance/system testing to improve robustness and efficiency of our model
- **Scalable Data Workflow**: Chunking in embedding process and custom dataset integration helps improve more detailed and data-specific results

Challenges:

- **Embeddings Constraint**: Need to efficiently embed large datasets

Next Steps and Sprint Retro

What Went Well:

- Switched LLM from LLaMa to ChatGPT
  - Reduced LLM response time from minutes to seconds
- Implemented embeddings chunking for vector search
- Improved and documented system testing by timing components

Next Steps:

- Allow the user to query a multimodal custom dataset
- Vector search ranking to improve response quality
- Further optimization of the RAG application
- Migrate this RAG application to a publicly accessible web app

Conclusion

Key Progress:

- Reduced response time
- System level optimizations
- Developing ability for a multimodal custom dataset

Thank You!