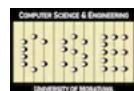




## — Challenge 01 —

Organized By

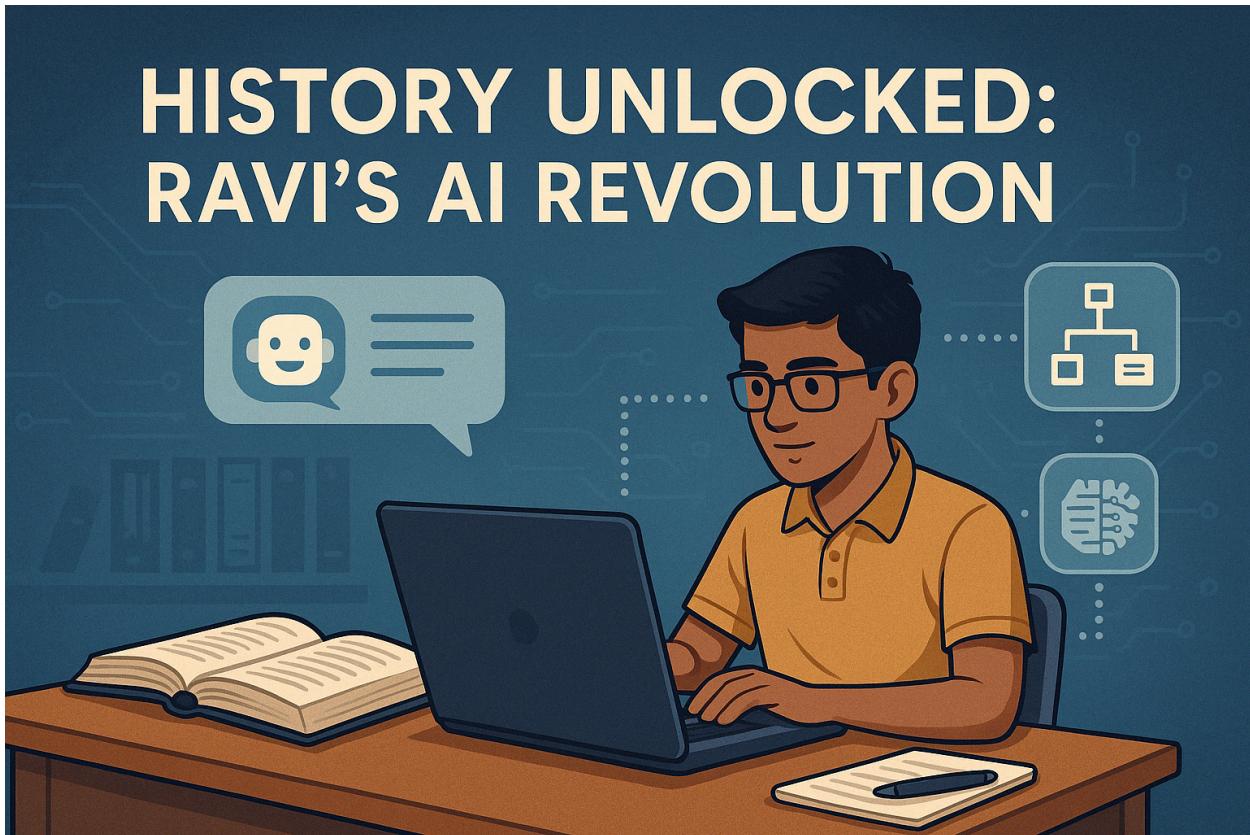


Department of Computer Science & Engineering  
University of Moratuwa

April 15, 2025

## 1. Background to the Problem

# History Unlocked: Ravi's AI Revolution



Meet **Ravi** - a bright and determined high-school student juggling classes, extracurricular activities, and the pressures of modern life. Every day, Ravi faces a daunting challenge: his history teacher assigns intricate questions based on the Grade 11 History textbook. With so many commitments on his plate, Ravi simply doesn't have the time to pore over every detail in the thick textbook. One afternoon, as Ravi struggles to balance his workload, inspiration strikes. What if he could create an application that could read and understand the entire textbook and then answer his teacher's questions on demand? Inspired by his need for quick and reliable information, he embarks on a journey to create an intelligent chatbot. This digital assistant harnesses the power of

advanced AI techniques to retrieve and synthesize key details from historical documents into clear, helpful responses. Throughout his adventure, Ravi refines his system by incorporating multiple collaborating agents—each with a distinct role—to emulate human-like decision-making and streamline the process of learning from a complex text.

## 2. Problem Statement

Design and develop a chatbot that consists of multiple agents collaborating to solve user queries about a given document. The chatbot should simulate intelligent decision-making by dividing responsibilities based on the nature of the task. At a minimum, it must include a Retrieval-Augmented Generation (RAG) agent capable of identifying and retrieving relevant information from the document and generating coherent, context-aware answers.

Participants are encouraged to expand the chatbot's functionality by adding additional agents such as task planners, summarizers, translators, sentiment analyzers, etc that work in concert to enhance its overall performance. Whether you choose a no-code solution using Langflow's drag-and-drop interface or decide to incorporate code for additional functionality, the primary goal is to create an innovative tool that mirrors Ravi's vision: transforming a vast amount of historical content into interactive, easily accessible knowledge.

By tackling this challenge, you will explore the intersection of education and technology, demonstrating how AI-powered multi-agent systems can revolutionize the way we access and understand historical knowledge.

### 3. Dataset

- Queries.json file containing queries.
  - Field Descriptions:
    - *ID*: The unique identifier for each question (note: we use 'ID' not 'query\_id')
    - *context*: The retrieved context from the textbook is used to generate the answer.
    - *answer*: The answer generated by your system.
    - *references*: A JSON-formatted string containing sections and pages, formatted as:

```
{"sections": ["section1", "section2"], "pages": ["page1", "page2"]}
```

- Textbook content in PDF format.
- Sample submission file showing expected format.

The test set with queries (without answers) and the required URLs, referring to web search agent resources, will be provided on the 20th of April.

### 4. Model Constraints

Participants are required to use only the **gemini-1.5-flash** model for this round. Using any other model will lead to disqualification.

Here are some helpful links to get started:

[Get a Gemini API Key](#)

[Gemini Documentation](#)

[Try the Gemini model in the Playground](#)

## 5. Vector Database

You're free to use any vector database that supports similarity search, including:

- Chroma
- Weaviate
- Qdrant
- Milvus
- FAISS
- Pinecone
- MongoDB with vector search

The choice of vector database should support:

- Efficient similarity search
- Storage of embeddings and metadata
- Fast query response times
- Easy integration with Python

Your performance may vary based on vector database selection and configuration. Ensure your chosen solution can handle the textbook embeddings within competition resource constraints.

The evaluation metric remains the same regardless of vector database choice.

## 6. Rules and Regulations

- Only use publicly available, free resources.
- Submissions must include all required columns in the specified format.
- Solutions must be reproducible and well-documented.
- You acknowledge and agree that the Future Minds organizing committee may, without any obligation to do so, remove or disqualify an individual or team, if Future Minds organizing committee believes that such individual or team, is in violation of these rules or violated the spirit of the competition or the platform in

any other way. The disqualifications are irrespective of your position on the leaderboard and completely at the discretion of the Future Minds organizing committee. Entry into this competition constitutes your acceptance of these official competition rules.

## 7. Additional Resources

- [RAG Tutorial](#)
- [RAG Langflow Implementation Tutorial](#)
- [AI Agents](#)
- [AI Agents Explained](#)
- [AI Agents Tutorial](#)

## 8. Deliverables

- A csv file containing the answers for the given queries.
  - Submission Format
    - For each ID in the test set, you must provide a complete response.  
The submission file should be a CSV with a header and have the following format:
      - ID, Context, Answer, Sections, Pages
- A JSON file representing the pipeline or workflow of the chatbot for those who are using Langflow. (<https://docs.langflow.org/concepts-flows>)
- A GitHub repository link of the implementation for those who are not using Langflow.
- A 15 minute Q&A will be held for the top selected teams before the award ceremony to determine the authenticity of the submissions.
- A video submission of no more than 15 minutes, including demonstrating answer generation for the test queries and a walkthrough of the implementation.
- (Optional) Any additional code or custom assets used in the project.

## 9. Evaluation Criteria

1. **Context Precision (20%)**: How relevant is the retrieved context to the question?
2. **Answer Faithfulness (20%)**: How well does the answer reflect the retrieved context?
3. **Answer Correctness (40%)**: How accurate is the answer compared to the ground truth?
4. **Reference Accuracy (10%)**: How precisely do the provided references match the relevant content?
5. **Innovation (10%)**: Evaluates how effectively Langflow and any supplementary tools or resources were leveraged to enhance the solution. This includes the strategic use of external components, APIs, and frameworks to improve the chatbot's capabilities. The integration of additional agents beyond the minimum requirement will also be considered under this criterion.