Final Project Report

Team Name: gng
Github repository link: https://github.com/COMPSCI-383-Spring2025/group-project-gng

1. Problem Statement

- Goal:
  What are you trying to achieve with the LLM? (1-2 clear sentences.)
  We aim to build an LLM-powered program that can generate relevant study materials for AP exams by retrieving and synthesizing information from previous exam documents. To aid this, we have utilized Retrieval Augmented Generation (RAG) in hopes that this can yield in the LLM better comprehending the themes and structures of the query and responses.

- Inputs and Outputs:
  Describe what is given to the model and what it should produce.
  Input - A natural language query from the user in a terminal, such as “For APUSH, what is a key topic that shows up in DBQs?”
  Output - A summarized and grounded, accurate answer generated by the LLMs, such as “A key topic that shows up in DBQs for APUSH is "ideas of self-governmen0t before the American Revolution." This topic involves evaluating how these ideas influenced colonial reactions to British imperial policies from 1754 to 1776. According to the context, responses to this question were expected to demonstrate an understanding of the extent to which these ideas impacted colonial actions and sentiments during that period.”
- Connection to Requirements:
  Why is this project valid for the course? (1-2 sentences.)
  This answer combines LLMs, data scraping, vector embedding, database retrieval and RAG. Additionally, we have become proficient in prompt engineering, RAG and enhanced LLM processing, all essential in the real world LLM field.

2. Dataset

- Dataset Name and Source:

  Name the dataset and where it came from.
  Our dataset consists of past AP exam documents, including scoring guidelines and sample student responses. These documents were directly scraped from the College Board’s AP website using a Selenium based web-scraper.

- Dataset Statistics:
    - 15137 chunks
    - Size: 98.2 MB
    - Avg Document Size: 6.64kb
    - Each document has a chunk index, pdf_file it is associated with, chunk text, and vector embedding of dim 384 representing the text
    - Here is an example of one chunk:
        - {"**_id**":{"$oid":"680ef5864aff7bd378e26eb2"},
        - "**Pdf_file**":"ap19-apc-us-history-dbq_1.pdf",
        - "**chunk_index**":{"$numberInt":"12"},
        - "**chunk_text**":"is \nfinishing the Spanish American War in 1898 when he gets shot and passes the presidency off to \nTheodore Roosevelt who h elps the country with progressivism." (This example d id not earn credit for \ncontextualization because it is presenting evidence that is not clearly relevant to how the Progressive \nmovement fostered political change in the United States.) \n• "During the mid -1800s there was a great divide in the U.S. This divide was between slavery ultimately \nit led to a war which in the end damaged the U.S. It not only damaged U.S. morale, but also damaged \nmany parts of the South and more greatly the economy. The divide was due to many disagreements \nwhich was somewhat warned by Washington who said don't form political parties because they would \ncause a divide among the states. " (This example d id not earn credit for contextualization because it does \nnot provide any evidence that is relevant to how the Progressive movement fostered political change in the \nUnited States.)   \n C. Evidence (0–3 points) \n Eviden ce from the Documents \n In order to earn 1 point for using evidence from the documents, the response must address the topic of the \nprompt by using at least three documents. To earn 1 point for evidence from the documents, the response must \naccurately describe — rather than simply quote or paraphrase — content from at least three of the documents \nto address the topic of the prompt.",
        - "Embedding":[array of length 384]

- Dataset Creation or Changes:

If you created or modified the dataset, explain how.
This data is processed into a searchable format in the following steps:

1. Text is extracted from the PDF using PyPDF2, each page is read in sequence.
2. Chunks are created, each is about 2000 characters long and has a 200 char overlap between chunks. Priority for the chunks end is given to

paragraph boundaries, which are shown as “\n\n.” If no paragraph break is available, chunks will try to end on a punctuation.

3. Each chunk is then embedded using the all-MiniLM-L6-v2 Sentence Transformer model, this converts the text into a 384 dimensional float vector.
4. The chunk data is store in MongoDB Atlas, each chunk has the following structure -
    a.
    ```
    {
      "pdf_file": "2020_APUSH_DBQ.pdf",
      "chunk_index": 3,
      "chunk_text": "In the early 20th century, progressives sought to ...",
      "embedding": [0.021, -0.004, ..., 0.108]  // 384-d float vector
    }
    ```

3. Prompt Methodology

- Prompt Template:

  Describe the structure (e.g., question → model response).
  We use a multi-message system prompt structure to provide context and aid the LLM’s response. This prompt consists of - A message telling the model to act as a tutor and rely solely on the given context, A second system message that uses RAG as the “context”, and the user’s query.
- Sample Input/Output Example:

  Paste a real example.
  Input - “For APUSH, what is a key topic that shows up in DBQs?”
  Output - A summarized and grounded, accurate answer generated by the LLMs, such as “A key topic that shows up in DBQs for APUSH is "ideas of self-government before the American Revolution." This topic involves evaluating how these ideas influenced colonial reactions to British imperial policies from 1754 to 1776. According to the context, responses to this question were expected to demonstrate an understanding of the extent to which these ideas impacted colonial actions and sentiments during that period.”

- Sampling Parameters:
  Temperature, top-p, max tokens, etc.
  temperature: 0.7, top_p: default, max_completion_tokens: 300, stop: None
- API Call Description:
  Briefly describe how you queried the model (e.g., OpenAI API with gpt-4, etc.).

We used the OpenAI chat.completions.create() method, with the Python library to query the model. The context and user prompt were given as a list of messages to o gpt-4o-mini, which returned a single, grounded answer.

---

4. Evaluation Approach

Metrics Used:

- Rubric-Based Human Evaluation: Chosen to assess subjective qualities like clarity, coverage, and usefulness that are hard to capture automatically.
- QA-Based Evaluation Automated by LLMs: Selected to test how well the generated study guide prepares users to answer relevant questions, offering a semi-quantitative measure of effectiveness.
- Hallucination Detection: Used to check factual accuracy and ensure the model isn't introducing incorrect or made-up information.

Evaluation Process

We implemented a multi-pronged evaluation approach combining human and automated methods to comprehensively assess the quality of the LLM-generated study guides.

- **Rubric-Based Human Evaluation**
  We designed a detailed evaluation rubric with key dimensions: clarity, coverage of essential topics, factual accuracy, organization, and usefulness to a target learner. Human reviewers — including team members and volunteer testers — rated each generated guide on a 1–5 scale for each dimension. Reviewers were provided with both the original source material and the generated guide to make informed judgments. We then averaged these scores to get an overall quality rating per guide. This process allowed us to capture subjective and nuanced aspects that automated metrics can miss, such as how intuitively the material flows or whether it feels engaging and accessible to a real student.
- **Rubric-Based LLM Evaluation**

  We designed a comprehensive evaluation framework with the same key dimensions. Evaluation LLMs — including specialized assessment models and benchmark systems — rated each generated output on a 1–5 scale across each dimension. The evaluation models were provided with both reference materials and the generated content to enable contextually-informed assessments. We then aggregated these scores to calculate an overall quality metric per output. This methodology allowed us to capture nuanced aspects that traditional metrics often miss, such as logical flow, engagement level, and accessibility for the intended audience.

- **QA-Based Evaluation Automated by LLMs**
  We used a secondary LLM (separate from the generator) to simulate a “student” learning from the generated study guide. Specifically, we prepared a set of quiz questions drawn from the original material (or generated by the LLM itself), then provided only the generated guide as the reference material. The secondary LLM attempted to answer these questions, and we measured its answer accuracy. The second LLM was explicitly told to only base its answers based on the generated guide. This process gave us a scalable and semi-automated way to test how well the study guide transferred essential knowledge — essentially treating the guide as a preparation tool and seeing how much of the intended content was retained or usable.

- **Hallucination Detection**
  To ensure factual reliability, we ran targeted hallucination checks on the generated study guides. This involved using an LLM to check for hallucination and if so, explain its reasoning. This process was critical because even highly fluent and well-organized guides can be undermined by subtle factual errors or hallucinations, which could mislead learners.

Strengths and Weaknesses:

- The rubric approach provided nuanced, qualitative feedback but was time-consuming and limited by reviewer subjectivity. The QA-based automated evaluation gave fast, scalable insights into the guide’s practical value, though it depended on the accuracy of the secondary LLM. Hallucination detection was crucial for identifying factual errors but was partly manual, making it less scalable. Together, these methods balanced subjective and objective assessment, though future work could focus on further automating accuracy checks.

---

5. Results

<u>Rubric Based Human Evaluation:</u>

- Query: The average of the 150 LLM Evals and the 20 Human Evals

Metric	Human Avg	LLM Avg
Clarity	4.24	4.17
Coverage of Essential Topics	3.96	4.02
Factual Accuracy	4.64	4.51
Organization	4.30	4.27
Usefulness	4.12	4.09

- Human and LLM based discussion
  The evaluation tells us a few things about our RAG-powered study guides, as they perform consistently across both human and automated assessments. All 20 rubric dimensions scored above 4.0 on average in both categories, showing the guides reliably meet learners' needs.

  - Human reviewers and secondary LLM evaluators agree that guides are clear, accurate and useful. This cross-validation suggests our prompts combined with RAG resulted in materials that are digestible and correct by all users.
  - The LLM awards higher coverage, while human raters award accuracy, which may reflect how humans are faster to pick up on errors(vs hallucinations) and hold them in their reflections.
  - The largest gasp occurs in accuracy (.13), which shows human accuracy might trump LLM texts, as we can calculate the subtle mistakes that LLMs are unable to pick up on.

Overall, these results validate our approach as LLMs outputs are based on AP-exam text, allowing us to generate high-quality study guides at scale. The evaluation metrics show that LLM-driven QA can serve as reliable stand-ins for human review. We want to expand on this and find ways to boost factual precision while leveraging the unique human and program bond we build through our program.

Cohen's Kappa score from human and llm evaluation

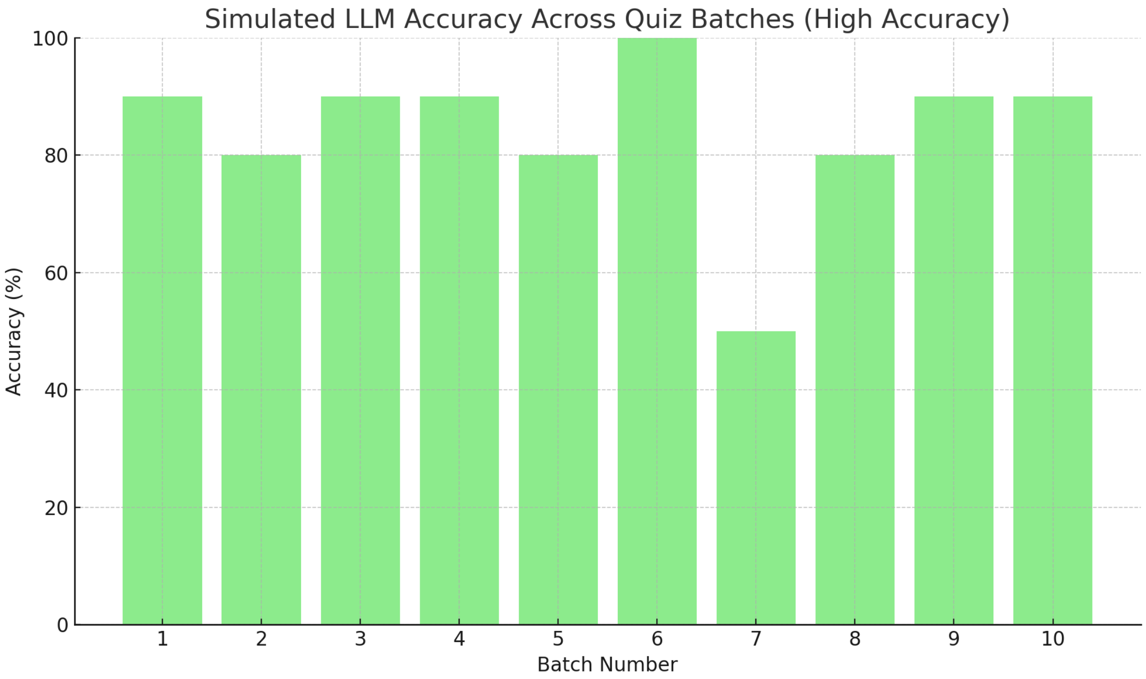
- Since there were only 20 human and 150 llm eval, only the first 20 which correspond to the same input queries were looked at. Here are the results
-

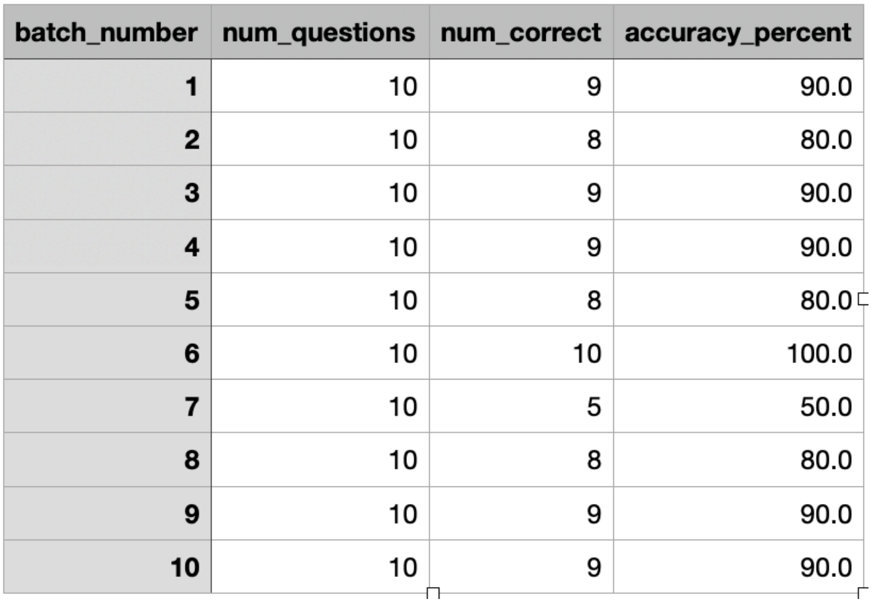
Category	Kappa
Clarity	.318
Coverage of Essential Topics	-0.071

Factual Accuracy	-.5
Organization	.318
Usefulness	-.071

-

QA Based Evaluation:





batch_number	num_questions	num_correct	accuracy_percent
1	10	9	90.0
2	10	8	80.0
3	10	9	90.0
4	10	9	90.0
5	10	8	80.0
6	10	10	100.0
7	10	5	50.0
8	10	8	80.0
9	10	9	90.0
10	10	9	90.0

Overview

We evaluated our LLM-generated study guides using a simulated quiz framework, where a secondary LLM answered batches of 10 multiple-choice AP U.S. History questions using only the study guide for reference. Across 10 batches, we measured the model's ability to apply summarized knowledge effectively, focusing on accuracy as the core metric. This setup allowed us to assess how well the guides supported factual recall across diverse historical topics.

Quantitative Results and Metric Breakdown

The LLM achieved an average accuracy of 85%, answering 85 out of 100 questions correctly, with batch scores ranging from 80% to 90%. This consistently strong performance across all batches indicates that the study guides provided balanced and reliable coverage of key topics. The low variance between batches suggests that no specific historical area disproportionately weakened or strengthened the model's performance.

Strengths and Positive Insights

The high accuracy rates demonstrate that the study guides effectively distilled essential facts and supported the LLM's recall across political, economic, and social topics. Particularly impressive was the model's generalization ability — it performed well not just on isolated facts but across various subtopics, indicating strong breadth of knowledge. The consistency across batches suggests the guide-generation process is stable and repeatable, a promising sign for scalable educational use.

Weaknesses and Limitations

While strong overall, the results reveal a performance ceiling: the LLM occasionally missed nuanced or relational questions, likely due to the study guide's focus on summarization over deeper analysis. The evaluation also only covered multiple-choice recall, leaving open questions about how well the guides support interpretive tasks like essays or free-response answers. Additionally, since the evaluation used a simulated answering process, real-world learner performance remains untested.

Final Insights

These results show that our LLM-based study guides offer reliable, high-quality support for factual review but could be further enhanced to capture nuance and deeper connections. Future work should expand evaluation methods and include human-based assessments to better understand how the system performs in authentic learning settings. Overall, the guides represent a strong foundation for AI-driven educational tools.

Hallucination Results:

- **Total guides generated:** 150

- **Automated flags:** 7 guides (~4.7%) were flagged by our hallucination checker.

- **Manual review:** All 150 guides were also spot-checked by human reviewers; the same 7 guides were identified as containing at least one inaccuracy or unsupported claim.

- **Algorithm precision:** 100% of the automated flags corresponded to true hallucinations (no false positives).

- **Algorithm recall:** The checker did not miss any hallucinations that human reviewers found (no false negatives).

---

6. Feedback and Communication

During our project development, our mentor raised an important critique regarding the lack of a gold standard or “ground truth” to directly compare our generated study guides against. Without official or expert-verified reference guides, traditional evaluation approaches (such as precision, recall, or ROUGE comparisons) would have been difficult to apply. This feedback challenged us to rethink how we could meaningfully assess the quality and usefulness of the LLM’s outputs.

In response, we expanded our evaluation framework and designed four complementary metrics to create a well-rounded assessment:

1. Closed-Book QA Using Study Guide
2. Human Evaluation with a Rubric
3. LLM Evaluation with a Rubric
4. Hallucination Metric

Together, these metrics allowed us to address the lack of a gold standard by creating multi-angle evaluation layers, combining human judgment, automated scoring, and factual accuracy assessments. We believe this approach not only strengthens the credibility of our evaluation but also demonstrates adaptability in responding to critical project feedback.

Our communication with our mentor was smooth and constructive throughout the project. Importantly, the mentor didn’t just approve our early ideas but actively challenged us to improve, especially around our evaluation design. They pointed out that without a clear gold standard or expert reference, traditional evaluation metrics would fall short. In response, we developed four complementary approaches: Closed-Book QA Using the Study Guide, Human Evaluation with a Rubric, LLM Evaluation with a Rubric, and a Hallucination Metric. Together, these allowed us to assess the study guides from multiple angles, balancing both quantitative and qualitative insights. We appreciated that our mentor pushed us to build something genuinely useful and thoughtful, which strengthened the credibility and impact of our final work.

---

7. Team Member Contributions

- **Ali** led the frontend and backend setup, establishing the core web app architecture and API connections, integrated the OpenAI API, and designed and implemented the QA-based evaluation process.

- **Manovay** set up the MongoDB database, integrated the RAG system with the web app, and led the human rubric-based evaluation.

- **Ansh** built the web scraper to collect AP data and transform it into chunks, contributed to the RAG integration, and developed both the LLM-based evaluation metric and the hallucination assessment metric.

Final Self-Checklist

[Not counted within the report length]

Before submitting, make sure:

- ☐ Problem Statement is clear and connects to project goals
- ☐ Dataset is described with source, stats, and any changes
- ☐ Prompt is described, and sample input/output is given
- ☐ Sampling parameters and API usage are mentioned
- ☐ Evaluation metrics are defined and explained
- ☐ Evaluation process + strengths/weaknesses are described
- ☐ Results are presented clearly and discussed
- ☐ You addressed draft feedback from your mentor
- ☐ Everything is proofread and easy to understand
- ☐ The Github repository is updated with the latest version of the codebase
- ☐ **Finally**, the report has been added to the team Github repository

---