Project Milestone 2

Hiranya Bandi Lasya Reddy Ednuri Samantha Michael
Uma Chavali

Table of contents

1 Questions for Ryan before project implementation

- Do we have to use Modal for our project?
- T&C

2 Goal of Project Milestone 2

Improve the initial prototype by integrating Retrieval Augmented Generation (RAG), implementing comprehensive testing, enhancing the overall project structure, and planning for user testing.

3 Project Requirements

3.1 1. PyTest Integration (15%)

- Implement at least 10 PyTest tests for LLM inference relevant to your use case
- Tests should cover various aspects of the application, including prompt handling, response generation, and error cases
- Provide options for running tests locally (via Ollama) or through an API (e.g., your modal endpoint)
- Include instructions in the README on how to run tests

3.2 2. GitHub Workflow Enhancement (15%)

- Develop at least one new feature for the application using GitHub Issues
- Create a Pull Request for the new feature, demonstrating proper Git workflow
- Document the process in the project write-up, including links to the Issue and PR

3.3 3. Retrieval Augmented Generation (RAG) Implementation (20%)

- Identify and integrate a suitable data source (e.g., CSV file, text corpus) for RAG
- Implement a basic RAG process using libraries like LangChain or similar tools
- One option, provide a Jupyter notebook demonstrating the RAG process (easier option)
- Ideally, integrate RAG directly into the application (e.g., via Streamlit)
- Consider these repos: RAG Techniques

3.4 4. Repository Organization and Best Practices (10%)

- Update the README with comprehensive project information, setup instructions, and usage guidelines
- Choose an appropriate project name and update all relevant files/directories
- Implement a clear and logical file organization structure
- Integrate development tools such as Black for code formatting, Bandit for security checks, and isort for import sorting
- Document the use of these tools in the README

3.5 5. Project Use Case Definition (15%)

In the write-up, include a detailed description of your LLM product:

- Product Overview
- Target User
- Problem Statement
- Key Features
- Scope Definition (In-scope and Out-of-scope features)
- Unique Value Proposition
- Success Metrics
- Potential Challenges
- Ethical Considerations

3.6 6. User Testing Plan (10%)

In the write-up, include a detailed plan for user testing:

- Target user group
- Recruitment strategy
- Testing methodology
- Feedback collection
- Metrics
- Timeline

3.7 7. Documentation and Reflection (15%)

In the write-up, include:

- A summary of implemented features and improvements
- Challenges encountered and how they were addressed
- Reflections on the RAG process and its impact on application performance
- Future directions and planned improvements for the next milestone
- Project use case definition (as detailed in section 5)
- User testing plan (as detailed in section 6)

4 Steps/Outline of our project

1. Define problem statement
2. Solidify idea - T&C
3. Collect Data and API
4. Use py scripts to collect data via links
5. Create a new repo
6. Create a new project/issues in GitHub, pull requests

7. Use RAG (our project should be RAG based anyways)

5 How to approve a Pull request in GitHub

1. Go to files changed
2. Click review changes
3. Leave a comment and click approve and submit review
4. Then it'll show up as merge pull request and just approve it

6 How to clean data

1. Go to website
2. Use jina r to convert text
3. Give it to chatgpt/google studio AI
4. Clean junk
5. Feed it to python script/model
6. RAG data Yay!

7 Output of our app

- Top 5 takeaways of T&C and do so in less than 10 words
- Company
- Make it similar to tosdr - has to do that or else too much text.

8 What to do for Project milestone 2

- Find T&C and prompt model to summarize and take model output and ground (TOSDR) output and run it through BERT (Deep Eval)
- Techniques for eval: BERT, ROUGE, BLEU
- Likert Scale - 1-5 scale
- Model metrics may perform poorly - DO NOT FEEL BAD! :)
- Make 1-5 examples to show the metrics and how our data/LLM is performing
- Quality over quantity!
- Qualitative analysis as a group of model outputs → use chatgpt as a judge → compare and contrast → now, trustable so yay!

9 References and Tools

- TOSDR
- Prompting for legal documents
- Eugeneyan
- Promptfoo
- Testing Packages for LLMs
- Langfuse
- Chunkviz
- RAG Techniques GitHub
- Evaluations and Metrics

10 Evaluation Notes

- Use Likert scale to rate the accuracy of the summary
- Evaluate: Fluency, Coherence, Consistency, Relevance, Toxicity
- Be aware of hallucination risk
- Use BERT Score for ground truth
- Implement Pairwise Cosine Similarity and Maximum Similarity
- Use ROUGE metric package (ROUGE-N, ROUGE 1, ROUGE 2)
- Be aware of BLEU Score limitations