MGMTMSA 440 Building LLM Powered Applications

Final Project

Multi-Agent LLMs For Wine Explorer

Reflective Essay

Group 2: Sen Yang, Cocoon Cao, Vicky Li, Darrel Wu

**Embedding Large Datasets:** Throughout the development of our wine recommendation system, we faced several challenges that helped us grow and learn as a team. One of the initial challenges we encountered was the slow process of embedding a large dataset of over 140,000 wine records. Embedding is a critical step where text data is transformed into numerical vectors, but attempting to do this all at once led to significant delays. Initially, we underestimated the time it would take and quickly realized that embedding such a large dataset in one go was inefficient.

To overcome this, we decided to test the code on a smaller subset of data first, allowing us to ensure that the core functions worked correctly before applying them to the entire dataset. This decision saved us from wasting time on potential bugs that would have been much harder to identify with the full dataset. While embedding the entire dataset still required patience, breaking the task into smaller batches allowed us to work more efficiently and progress with confidence. This experience taught us a valuable lesson: for large computational tasks, breaking them into manageable parts can significantly improve workflow efficiency. Had we taken this approach from the start, we would have saved valuable time.

**Managing Multi-Agent Systems:** Another significant challenge was managing a multi-agent system, where different agents handled tasks like research, summarization, and social media content generation. We quickly realized that coordinating these agents effectively required more than just assigning tasks. Each agent needed a clearly defined role, and we had to ensure they worked together without overlap or confusion. Much time was spent adjusting and fine-tuning the system to achieve a balanced workflow between the agents. In hindsight, creating a clearer plan for how the agents would interact and work together at the outset could have streamlined the process and reduced the amount of tweaking needed later.

**Key Lessons Learned:** Through these challenges, we gained several key insights:

1. **Breaking Complex Problems into Smaller Steps:** Tackling large tasks, such as embedding a massive dataset, is more manageable when broken down into smaller, testable steps. This approach prevents frustration and saves time, especially when dealing with large datasets or resource-intensive tasks.
2. **Prompt Engineering and AI Agents:** When working with AI agents, thoughtful and clear instructions are crucial. We learned that spending time upfront on refining the prompts for these agents drastically improved the quality of their outputs. The better the prompts, the better the agents performed their tasks.
3. **System Coordination:** Building a system with multiple AI agents requires careful planning and coordination. Ensuring each agent works harmoniously with others is essential to prevent inefficiencies and miscommunication. We learned that investing time early in designing the flow of interactions between agents would have prevented many of the issues we later encountered.

**What We Would Do Differently:** If we were to approach a similar project again, we would focus more on **initial planning**. For example:

* We would create a more detailed process for splitting tasks to optimize our workflow from the beginning.
* We would spend more time **refining prompts for AI agents** before running them on large datasets, to save time spent troubleshooting.
* Finally, we would ensure that the entire **multi-agent system** is well-coordinated before deployment, preventing many of the technical issues we encountered during this project.

By implementing these changes, the process would have been more efficient, allowing us to focus on improving the recommendation system rather than spending time troubleshooting technical issues.