CSAI 422: Advanced Topics in Generative AI

Group Project: SynthesisTalk - Collaborative Research Assistant

Overview

In this group project, you will apply the concepts learned throughout this course to build a sophisticated LLM-powered application that demonstrates conversational agency, tool usage, and advanced reasoning techniques. Working in teams of three, you will create "SynthesisTalk," an intelligent research assistant that helps users explore complex topics through an interactive, conversational interface.

SynthesisTalk combines document analysis, web search capabilities, and customizable outputs to support deep research on complex topics. The system will demonstrate how LLM-based conversational agents can effectively utilize tools, maintain context, apply reasoning techniques, and generate useful outputs in various formats.

Learning Objectives

- Implement conversational agency concepts including tool usage and reasoning techniques
- Design and build effective LLM workflows for complex tasks
- Create a responsive frontend that effectively communicates with LLM-powered backends
- Develop practical experience with modern AI application development
- Apply collaborative software development practices in an AI context

Project Requirements

Core Features

1. **Contextual Research Conversation**
   - Implement multi-turn conversations that maintain context about the research topic
   - Support document uploads (PDFs, text files) with content extraction and analysis
   - Integrate web search to supplement user knowledge and document content
   - Maintain coherent conversation across multiple information sources
2. **Intelligent Synthesis Engine**
   - Extract key information from documents and web searches
   - Connect related concepts across different sources
   - Generate insights based on patterns in collected information
   - Implement at least one advanced reasoning technique (Chain of Thought or ReAct)

3. **Flexible Output Generation**
   - Generate structured summaries in at least two different formats
   - Create simple visualizations based on research findings
   - Support exporting findings in a standard document format
4. **Tool-Enhanced Experience**
   - Implement at least four different tool types:
     - Document analysis tools (extraction, summarization)
     - Web search tools for fact-finding and verification
     - Note-taking and organization tools
     - Explanation and clarification tools

Technical Requirements

1. **Frontend**

- Build a React-based web application with responsive design
- Implement a chat interface supporting text and document uploads
- Create components for displaying research insights and visualizations
- Design a context management UI showing current research topics and sources
- Provide real-time feedback for lengthy operations

2. **Backend**

- Develop a FastAPI or Flask backend serving LLM integration
- Implement a tool management system for handling different LLM tool calls
- Create a document processing pipeline for extracting and indexing content
- Design a conversation history management system
- Implement at least one advanced reasoning technique (Chain of Thought, ReAct)

3. **LLM Integration**

- Integrate with an LLM API (Gemini, Groq, NGU LLM)
- Define and implement tools for various research functions
- Create a workflow that effectively orchestrates multiple tools
- Implement some form of self-correction mechanism

Team Structure and Roles

Each team of three students should distribute responsibilities across these core areas:

- Frontend development and UX design
- Backend API and tool integration
- LLM workflows and reasoning implementation

While each team member will have primary responsibility for one area, all members should collaborate across components to ensure effective integration.

Development Approach

You are encouraged to use Aider (or similar AI-assisted development tools) to help with implementation. This project is designed to give you hands-on experience with both building AI systems and using AI tools in the development process.

Deliverables

1. **Working Prototype**
   - Functional SynthesisTalk application with all core features
   - Deployed version or clear instructions for local setup
2. **Source Code**
   - Well-organized GitHub repository with documentation
   - You must have REGULAR commits throughout the project duration, showing the contribution of different team members, no "big bangs"!
   - README with setup instructions
   - Comments explaining key implementation decisions
3. **Technical Report (5-8 pages)**
   - System architecture description
   - Implementation details for reasoning techniques
   - Description of tool integration approach
   - Discussion of challenges encountered and solutions
   - Evaluation of system performance
4. **Presentation**
   - 15-minute team presentation
   - Live demonstration of key features
   - Discussion of technical implementation
   - Reflection on development process and lessons learned

Evaluation Criteria

Your project will be evaluated based on the following criteria:

1. **Functionality (40%)**
   - Successful implementation of all core features
   - Effectiveness of the conversation and research experience
   - Quality and usefulness of generated outputs
   - Robustness and error handling
2. **Technical Implementation (30%)**
   - Quality of code and architecture
   - Effective use of LLM capabilities
   - Implementation of reasoning techniques
   - Integration of frontend and backend components
3. **User Experience (15%)**
   - Intuitive interface design
   - Responsiveness and performance

- Clarity of information presentation
- Overall usability

4. **Documentation and Presentation (15%)**
   - Quality of technical report
   - Clarity of presentation
   - Completeness of documentation
   - Reflection on development process

Important Notes

- You may use any LLM provider, but your application should be designed to be easily adapted to different providers
- You should implement proper error handling and fallback mechanisms
- Consider usability and accessibility in your design
- Document any limitations or potential improvements for your system