Technical Documentation

Chemistry Chatbot Agent

Author: Xi Chen

Date: July 2025

1. Architecture Overview

This Chemistry Chatbot Agent is designed to assist users in querying and analyzing chemical information using natural language inputs. The system is built using Streamlit for the user interface and integrates OpenAI's GPT-based language model as the agent core. It uses function calling to interact with multiple chemistry-specific APIs. The system is modular, enabling easy expansion of capabilities by adding new tools or APIs.

Components include:
- Chat Interface (Streamlit)
- Chat Manager and Memory Modules for maintaining conversation context
- Toolset for PubChem, RDKit, PDB, SmallWorld, Web Search, and Patent APIs
- Function routing logic to delegate tasks to the appropriate tool

2. API Integration Notes

- PubChem API: Used for retrieving molecular structure and SMILES information based on compound name.
- RDKit: Used locally to compute molecular descriptors such as MW, TPSA, and logP based on SMILES.
- PDB API: Used to query protein metadata (resolution, species, expression system, etc.) using a given PDB ID.
- SmallWorld API: Submits molecular structure to find similar compounds, with similarity scores.
- Web Search: General chemistry queries are routed to a search tool for internet-based results.
- Patent API (Bonus): Supports Google Patent queries using SerpAPI, returning title, authors, and filing status of relevant patents.

3. Design Decisions

The system was developed using Streamlit to enable rapid prototyping and intuitive deployment. OpenAI's function calling was selected for natural multi-turn conversation flow with embedded tool usage. The system maintains short-term memory (last 4 message pairs) to support follow-up questions. Each API integration is modularized in the `tools/` directory, enabling fast iteration and future extensibility. An additional design decision was the use of a planner-based agent, enabling optimal tool selection paths to minimize API calls and reduce response time. Proper prompts and data filters are designed to filter out hallucinations when extracting knowledge in chat history.

4. Known Limitations

- Currently support searches of one chemical at a time.

- Rate-limiting for external APIs (PubChem, SerpAPI) is minimally handled and can be improved and replaced with downloaded packages.

- The system is not yet optimized for long-term session memory or multi-user sessions.

5. Technical Requirement Summary

✓ = Implemented ✗= Not Fully Implemented

- **Core Requirements:**
  - ✓ Molecular Structure Lookup (PubChem API)
  - ✓ Molecular Property Calculations (RDKit)
  - ✓ Protein Structure Information (PDB Database)
  - ✓ Chemical Similarity Search (SmallWorld API)
  - ✓ Web Search Integration (DuckDuckGo)
  - ✓ BONUS: Patent API Integration (Google Patents via scraping)
- **Technical Requirements:**
  - ✓ **Framework Choice:** Streamlit was selected for its simplicity and ability to natively render chemical images and charts. The system uses the OpenAI SDK with a function-calling agent framework, allowing seamless tool invocation and easy integration of future capabilities.
  - ✓ **Session Management:** Session state is preserved through Streamlit's built-in state management, allowing the chatbot to retain message history across interactions. This supports a smooth user experience with multi-step tasks.
  - ✓ **Memory:** The system maintains contextual memory of the last 4 message pairs. It does more than recall: it extracts key variables such as molecule names and SMILES from prior conversations and combines them with a planner mechanism to dynamically select the shortest tool path to compute requested properties. This minimizes redundant API calls and improves overall system efficiency and response time.
  - ✓ **Error Handling:** Each tool returns structured responses with success/failure flags and descriptive error messages. The system handles API exceptions, invalid inputs, and network failures gracefully. GPT-4o is prompted to interpret ambiguous, misspelled, or non-chemical queries with informative responses.
  - ✓ **Rate Limiting:** Basic delays (sleep) included for the openAI API to avoid rate limit (can be further improved for other APIs.
  - ✓ **Response Processing:** Raw API outputs are parsed and reformatted into conversational replies using GPT-4o. The system delivers clean, user-friendly summaries, often combining numerical results, chemical metadata, and molecular visuals to support interpretability.