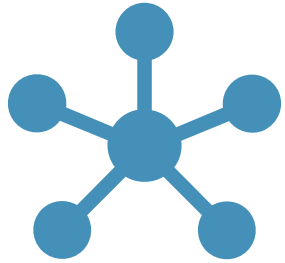


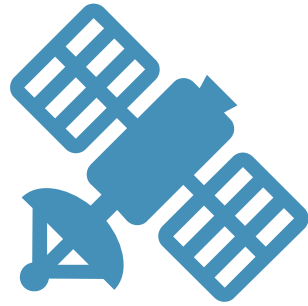


CMD-ASSISTANT

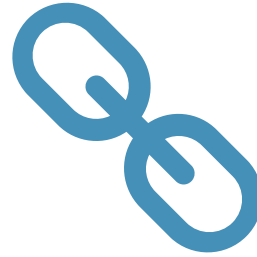
MADE BY JIAFANG BIAN ZIMO LI HAOSENG YANG



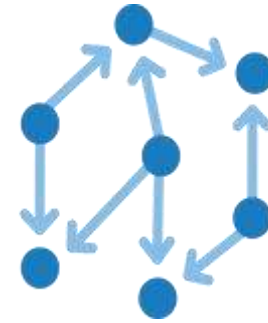
Application Background and Purpose



Function Overview



Technical Architecture



Future Plans



APPLICATION BACKGROUND AND PURPOSE



- **Application Background:**
In modern operating systems, users often need to perform various file operations. Traditional command-line operations may not be user-friendly enough for the average user.
- **Application Purpose:**
To provide an intelligent command assistant that simplifies the file operation process. By utilizing natural language processing technology, we aim to reduce the learning curve for users. Additionally, leveraging LLM (Large Language Model) agents will enhance the user interaction experience.

FUNCTION OVERVIEW



■ Main Features:

- **File Search:** Search for files based on user input.
 - Supports both fuzzy matching and exact matching.
- **Command Generation:** Automatically generate corresponding Windows commands.
 - Includes syntax checking and error prompts.
- **History Record:** Keep a record of user operation history.
 - Allows users to view and reuse previous commands easily.
- **Intelligent Dialogue:** Utilize the LLM agent for natural language understanding and generation.
 - ◦

TECHNICAL ARCHITECTURE



■ Tech Stack:

- **Python:** The primary programming language.
- **Requests:** Used for interacting with AI models.
- **Pickle:** Used to save and load user operation history.
- **Win32com:** Handles Windows-specific operations.
- **LLM Agent:** Processes natural language input and generates responses.

■ Workflow:

1. User Input →
2. Determine if a search is needed →
3. Execute file search →
4. Generate command →
5. Execute command.

- **Feedback Mechanism:** Provides feedback information on command execution.

- **LLM Interaction:** Understands user intent through the LLM agent and generates corresponding commands.

FUTURE PLANS



- **Function expansion:**
Increase support for other operating systems (such as Linux, MacOS).
Add more complex file operation commands, such as batch renaming, file compression, etc.
- **User interface improvements:**
Provide a graphical user interface (GUI).
Enhance user interaction experience, supporting voice input and output.
- **LLM capability enhancement:**
Continuously optimize the accuracy of LLM agent responses and context understanding.
Collect user feedback to fine-tune natural language processing models.



THANKS!