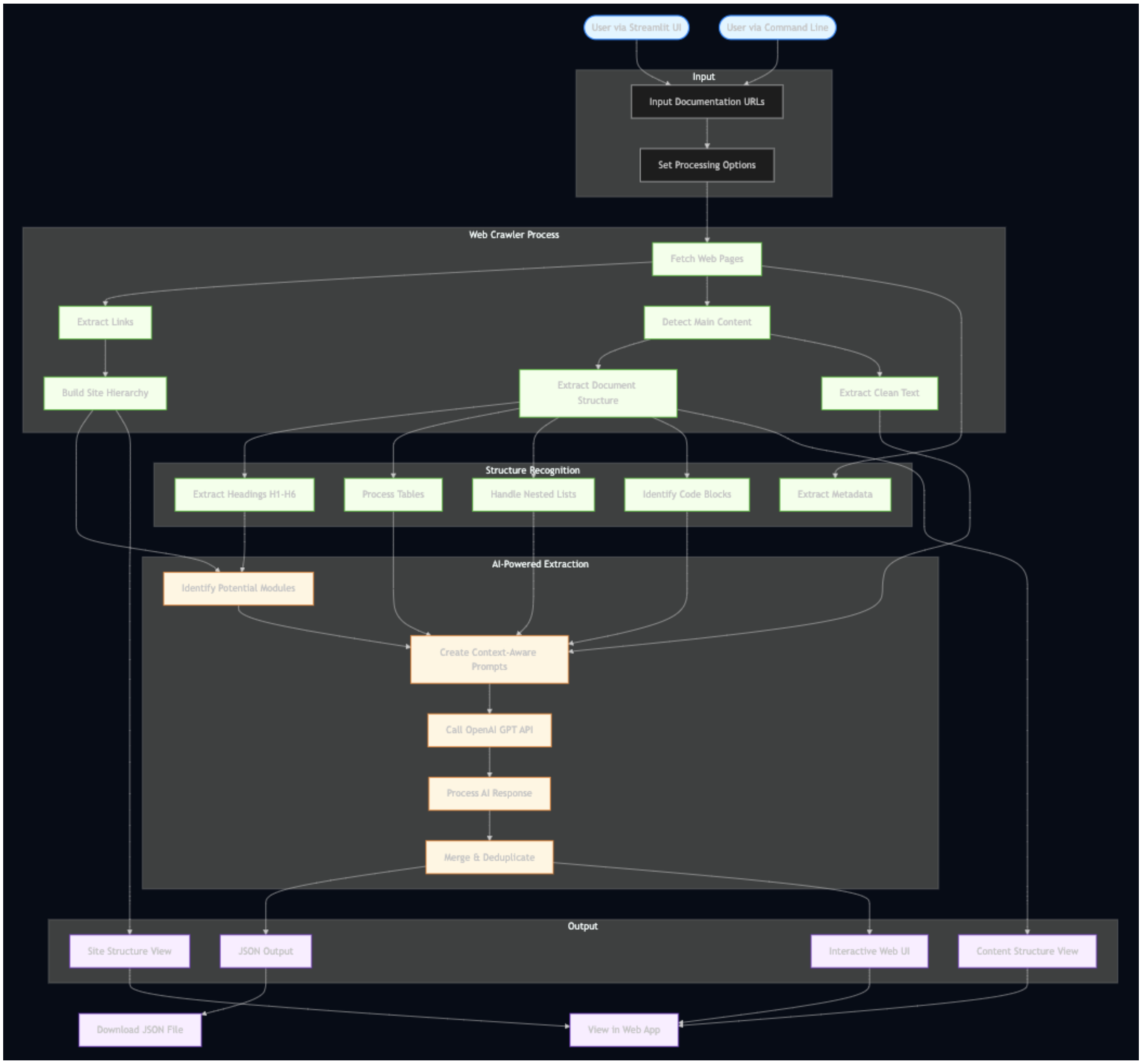
Pulse: AI-Powered Analyzer Documentation

Project Overview

Pulse is an intelligent documentation analysis tool that extracts structured information from help and documentation websites. It automatically identifies modules, submodules, and generates detailed descriptions by crawling websites and leveraging OpenAI's language models for content understanding.

Project Flow



Process Explanation

- **User Input**: Users can access Pulse through either the Streamlit web app or command-line interface
- **Configuration**: Set crawling parameters like depth, page limits, and model selection
- **Web Crawling**: The system fetches web pages and extracts links while building a hierarchical representation
- **Content Extraction**: Advanced algorithms detect main content areas while filtering out irrelevant elements

- **Structure Recognition**: The system identifies and extracts structural elements like headings, tables, lists, and code blocks
- **AI Analysis**: OpenAI's language models analyze the content to identify modules and submodules
- **Result Generation**: The system organizes the extracted information into a structured format
- **Output Presentation**: Results are presented through interactive interfaces or exported as JSON

Project Structure

The Pulse project has a clean, modular organization:

```
pulse/
├── app/
│   └── app.py            # Streamlit user interface
├── scripts/
│   └── cli.py            # Command-line interface
├── utils/
│   ├── crawler.py        # Web crawling functionality
│   └── extractor.py       # Module extraction with OpenAI
├── requirements.txt      # Dependencies
└──. env                  # Environment variables (for API keys)
```

Core Components

1. Crawler (utils/crawler.py)

- Handles web crawling and content extraction
- Implements HTML structure preservation (tables, lists, headings)
- Extracts document metadata and hierarchy
- Uses advanced selectors for main content detection
- Maintains a hierarchical structure of content

2. Extractor (utils/extractor.py)

- Processes crawled content to identify modules and submodules
- Uses OpenAI API (GPT-3.5/GPT-4) for extraction
- Implements content chunking to handle large documents
- Creates specialized prompts based on document structure
- Merges and deduplicates extracted information

3. Streamlit App (app/app.py)

- User-friendly web interface
- Input validation and options configuration
- Visual progress tracking during extraction
- Multiple views for results (interactive, JSON, structure)
- Visualizes document metadata and structure

4. CLI Tool (scripts/cli.py)

- Command-line interface for batch processing
- Options for output formats and additional data
- Logging and error handling
- Useful for automation and integration

Technologies Used

Web Crawling & Parsing

- **requests**: HTTP requests to fetch web pages
- **BeautifulSoup4**: HTML parsing and content extraction
- **trafilatura**: Advanced text extraction from web pages
- **html2text**: Converts HTML to markdown while preserving structure

Natural Language Processing & AI

- **OpenAI API**: Powers module/submodule identification
- **GPT-3.5/GPT-4**: Models used for content understanding
- **Regular expressions**: Pattern matching for structure recognition

User Interface

- **Streamlit**: Interactive web application framework
- **Command-line interface**: Terminal-based access

Data Processing

- **JSON**: Data storage and exchange format
- **Python collections**: Efficient data structures (defaultdict)
- **Content chunking**: Handles large documents efficiently

Project Infrastructure

- **python-dotenv**: Environment variable management
- **logging**: Application logging and debugging
- **argparse**: Command-line argument parsing

Running the Application

Prerequisites

- Python 3.8+
- OpenAI API key

Setting Up

Clone the repository
Install dependencies: pip install -r requirements.txt
Create a .env file with your OpenAI API key
Run the application through either interface:
Streamlit: streamlit run pulse/app/app.py
    CLI: python pulse/scripts/cli.py --urls

Conclusion

Pulse demonstrates how modern AI can be combined with web crawling technologies to extract and structure information from documentation websites. The modular design allows for easy extension and adaptation to various documentation formats and structures.