Web Research Agent - Documentation

Overview

The Smart Web Research Agent is an AI-powered tool designed to autonomously perform web research. Given a user query, it performs keyword extraction, searches the web, scrapes relevant content, analyzes and summarizes the content, and returns a concise and informative answer.

Built using Python, LangChain, DuckDuckGo Search, BeautifulSoup, and Groq's LLaMA-3 model, it offers both a CLI and a Streamlit-based web interface.

System Architecture

Components & Workflow:

1. **Query Analyzer:**
   Extracts key terms from the user's natural language query using an LLM.
   Converts questions into precise search terms.

2. **Search Tool (DuckDuckGo):**
   Performs a web search using the extracted keywords.
   Retrieves the top 3 relevant web page URLs.

3. **Scraper Tool (BeautifulSoup):**
   Visits each URL and extracts clean text content from the web pages.
   Cleans out scripts, ads, navigation elements.

4. **Content Analyzer (LLM):**
   Summarizes each page into concise, easy-to-read formats.
   Identifies key insights, facts, and perspectives.

5. **Synthesizer (LLM):**
   Merges individual page summaries into a single, unified response.
   Removes duplicates and maintains a coherent narrative.

File Structure

```
web-research-agent/
├── app.py            # Streamlit frontend
├── main.py           # CLI version
├── agent.py          # Core agent logic and orchestration
├── tools.py          # All tool logic including search, scraping, summarizing
├── requirements.txt   # Required Python libraries
├── .env              # Environment variables (API keys)
```

Example Use Case

Input:

> "What are the latest developments in quantum computing?"

Process:

- Extracts keywords: "latest quantum computing developments"
- Searches web and retrieves articles from sources like Nature, MIT Tech Review
- Scrapes readable parts of these pages
- Summarizes the core points
- Synthesizes them into a 3-4 paragraph response

Output: A concise summary of major trends, breakthroughs, and current debates in quantum computing.

Tech Stack

Component	Technology Used
Language Model	Groq API with LLaMA-3
Web Search	DuckDuckGo Search
Web Scraping	BeautifulSoup
App Interfaces	CLI (Python), Streamlit (UI)

Prompt Design

- **Keyword Extraction:**
  Extract the most relevant search keywords from: "{user_query}".

- **Content Summarization:**
  Summarize the following webpage content into key points.

- **Synthesis:**
  Merge the following summaries into a single coherent answer.

Connecting to External Tools

- **Groq API**:
  Used for all LLM completions (keyword extraction, summarization, synthesis).
  Connected via `groq` Python SDK and secured using `.env`.

- **DuckDuckGo Search**:
  Uses `duckduckgo_search` Python package to retrieve URLs.
  No API key required, lightweight and fast.

- **Web Scraping**:
  Implemented using `requests` and `BeautifulSoup`.

Error Handling

- Invalid or unresponsive URLs are skipped with logs.
- Content-less pages are ignored.
- LLM failure fallback mechanisms are present.
