

Research Assistant Agent Project Report

Group Name: Lionel DeLuna

Participants: Lionel DeLuna

July 22, 2025

1 Project Overview

This project develops a Research Assistant Agent designed to assist users in researching topics by retrieving information from web sources, evaluating their credibility, summarizing key points, and generating structured reports with proper citations. The source code for this project is available at [GitHub](#). The agent is implemented in a Google Colab environment, leveraging web search and document processing tools to produce organized research outputs for academic or professional use. Its purpose is to streamline the research process, making it easier for students and professionals to gather and synthesize information efficiently.

2 System Architecture

The agent follows a modular architecture with the following components:

- **Input Processing:** Parses user queries to identify the topic and research scope, with validation to reject inappropriate or harmful inputs.
- **Memory System:** Stores retrieved sources, summaries, citations, and feedback logs in a structured dictionary for easy retrieval and management.
- **Reasoning Component:** Employs Chain-of-Thought reasoning to evaluate source credibility and plan the structure of the research report.
- **Output Generation:** Produces a formatted Word document containing organized findings, summaries, and citations using the python-docx library.

The agent integrates two external tools: googlesearch-python for retrieving web sources and python-docx for generating reports. A reinforcement learning component adjusts source selection weights based on user feedback, improving decision-making over time. These components interact seamlessly to process user inputs, retrieve and evaluate data, and generate structured outputs.

3 Implementation Details

Key technical decisions and approaches include:

- **Web Search Tool:** Selected googlesearch-python for its simplicity and compatibility with Google Colab, enabling efficient source retrieval.
- **Credibility Scoring:** Implemented a heuristic-based scoring system that evaluates sources based on domain type (e.g., .edu, .gov) and content length, prioritizing reliable sources.
- **Summarization:** Used a simple approach of extracting the first two sentences and top keywords to create concise summaries, balancing quality and resource constraints.
- **Resource Optimization:** Limited source retrieval to five sources and content to 1000 characters per source to manage Colab's computational limits.
- **Report Generation:** Utilized python-docx to create structured Word documents, ensuring accessibility for users without additional software.
- **Error Handling:** Added robust error handling for web requests and fallback to cached sources to ensure reliability.
- **Reinforcement Learning:** Incorporated a feedback mechanism to update source selection weights based on user ratings, enhancing future performance.

The Chain-of-Thought reasoning pattern guides the agent in evaluating sources and structuring reports, while safety measures like input validation prevent inappropriate queries.

4 Evaluation Results

The agent was tested with queries such as "artificial intelligence ethics" and "climate change impacts." Testing involved:

- **Source Retrieval:** The agent retrieved 3-5 sources per query, with .edu and .gov domains prioritized due to higher credibility scores (0.7-1.0).
- **Summarization:** Summaries captured key points and keywords, though limited by the simplistic truncation approach.
- **Report Generation:** Word documents were generated with clear headings, summaries, credibility scores, and citations, meeting academic formatting standards.
- **Reinforcement Learning:** Simulated user feedback (positive: 1.0, negative: -1.0) adjusted source weights, improving selection in subsequent runs.

Tests were conducted with small samples to verify stability before scaling to broader queries. The agent consistently produced usable reports, though summarization quality varied with content complexity.

5 Challenges and Solutions

Several challenges arose during development:

- **Challenge:** Web search failures due to network issues or blocked requests. **Solution:** Implemented try-except blocks and fallback to cached sources to ensure continuity.
- **Challenge:** Simplistic summarization missing nuanced content. **Solution:** Added keyword extraction to highlight key themes, improving summary relevance.
- **Challenge:** Resource constraints in Google Colab limiting processing power.

Solution: Optimized by capping source count and content length, ensuring efficient execution.

- **Challenge:** Limited citation formatting options. **Solution:** Focused on basic numbered citations, with plans for future expansion to standard formats like APA.

These solutions ensured the agent remained functional and reliable within the project scope.

6 Lessons Learned

The project provided valuable insights:

- **Modular Design:** Breaking the agent into components (input, memory, reasoning, output) simplified development and debugging.
- **Error Handling:** Robust error handling for external tools was critical for reliability, especially in unpredictable web environments.
- **Reinforcement Learning:** Even simple feedback mechanisms significantly improved agent performance over time.
- **Resource Management:** Optimizing for Colab's constraints taught the importance of balancing functionality with computational limits.

These lessons underscore the value of iterative development and testing in constrained environments.

7 Future Improvements

With additional time, the agent could be enhanced by:

- **Advanced Summarization:** Integrating NLP models like BERT for more accurate and nuanced summaries.

- **Vector Databases:** Using tools like FAISS for faster and more efficient knowledge retrieval.
- **Citation Flexibility:** Supporting multiple citation formats (e.g., APA, MLA) to meet diverse academic needs.
- **User Interface:** Developing an interactive interface for real-time user feedback and query refinement.
- **Scalability:** Expanding the agent to handle larger datasets and more complex queries without performance degradation.

These improvements would enhance the agent's utility and adaptability for broader research applications.