**Virtual Research Assistant Agent – “AskAcademia”**

Saima Sano

ITAI 2377

Professor Anna Devarakonda

Date: 07/23/2025

**Problem Statement**

Academic research can be lengthy and daunting, particularly for beginners unfamiliar with academic databases, citation formats, or evaluating credible sources. With an abundance information available online, users find it challenging to wade through relevant, trustworthy sources and condense them into coherent research.  
My agent solves this issue by being a Virtual Research Assistant that can:  
- Retrieve and summarize relevant information from academic and credible sources.  
- Evaluate the quality and relevance of each source.  
- Synthesize information into a coherent report.  
- Organize sources and generate citations in standard formats.

**Project Option**

I have selected Option 1: Research Assistant Agent.

**Agent Design**

The agent will follow modular architecture comprising the following components:

1. **User Interface (UI):** Command-line input for entering research topics.
2. **Information Retrieval Module:** Queries academic databases (e.g., Semantic Scholar, PubMed), DuckDuckGo Search Engine and Wikipedia
3. **Summarization Module:** Uses Hugging Face Transformers to summarize content.
4. **Source Evaluation Module:** Scores sources based on credibility indicators.
5. **Organization Module:** Structures summaries into a formatted output.
6. **Citation Generator:** Formats references using APA style with Python datetime module.

**Workflow:**

User Input → Retrieval → Summarization → Structuring → Citation → Final Report

**Tool Selection:**

I am using free and open-source tools to build the research assistant:

* **DuckDuckGo-search:** For retrieving web-based search results.
* **Wikipedia:** For concise summaries of academic topics.
* **Semantic Scholar API:** provides access to peer-reviewed research papers and metadata (title, abstract, authors, journal)
* **CORE.ac.uk API:** offers millions of open-access research papers for academic purposes.
* **Hugging Face Transformers (e.g., distilbart-cnn-12-6):** For local or hosted text summarization.
* **datetime:** For generating current-year APA-style citations.
* **Python:** The core programming language for implementation.

**Development Plan**

* **Day 1:** Setup Environment & Define Scope  
  Install required packages and test Hugging Face summarizer.
* **Day 2:** Build Search & Wikipedia Modules  
  Implement retrieval logic for DuckDuckGo and Wikipedia.
* **Day 3:** Summarization Pipeline  
  Summarize results using distilbart via Hugging Face.
* **Day 4:** Citation & Structuring Module  
  Format citations and organize outputs.
* **Day 5:** Error Handling & Feedback Logging  
  Add error handling and log user ratings.
* **Day 6:** Polish Output & Optional Export  
  Format for readability; optional export to Word.
* **Day 7:** Final Testing & Report Completion  
  Perform full tests and finalize documentation.

**Evaluation Strategy**

I will evaluate the success of the virtual research assistant based on the following metrics:

1. Accuracy: Are summaries and citations, correct?
2. Relevance: Are sources aligned with the topic?
3. User Experience: Is the user interface intuitive and easy to use?
4. Efficiency: Can the agent complete a task in under 5 minutes?

**Resource Requirements**

* Google Colab / Jupyter Notebook for development.
* Or Local machine with Python installed.
* Internet access for API-based searches (DuckDuckGo, Wikipedia).
* Hugging Face hosted or local models (free tier).
* Text file storage for user feedback logs.

**Risk Assessment**

|  |  |  |
| --- | --- | --- |
| Risk | Impact | Mitigation |
| API Rate Limits | Low | Use local caching or retry logic |
| Inaccurate Summaries | Medium | Tune summarization prompts or switch models |
| Source Irrelevance | Medium | Manually filter or rerun search with refined query |
| Technical Bugs | Medium | Add try/except and logging for traceability |