

AI System to Automatically Review and Summarize Research Papers

1. Introduction / Objective

This system automates the process of creating systematic reviews of academic papers through a structured workflow. It handles everything from initial paper search to final draft generation using a directed graph architecture.

2. Methodology / Workflow

1. Research Phase

- Topic planning and scoping
- Automated paper search via Semantic Scholar
- Smart paper selection (up to 3 papers – adjustable)
- Automatic PDF retrieval

2. Analysis Phase

- PDF text extraction
- Section-by-section analysis
- Key finding identification
- Cross-paper comparison

3. Writing Phase

- Automated section generation
- Abstract (100-word limit)
- Methods comparison
- Results synthesis
- APA reference formatting

4. Review Phase

- Quality assessment
- Revision suggestions
- Additional research triggers
- Final draft preparation

3. Modules

Paper Retrieval Module: Handles topic-based academic paper search and PDF download using Semantic Scholar API.

Text Extraction Module: Extracts and processes text from PDF documents for structural segmentation.

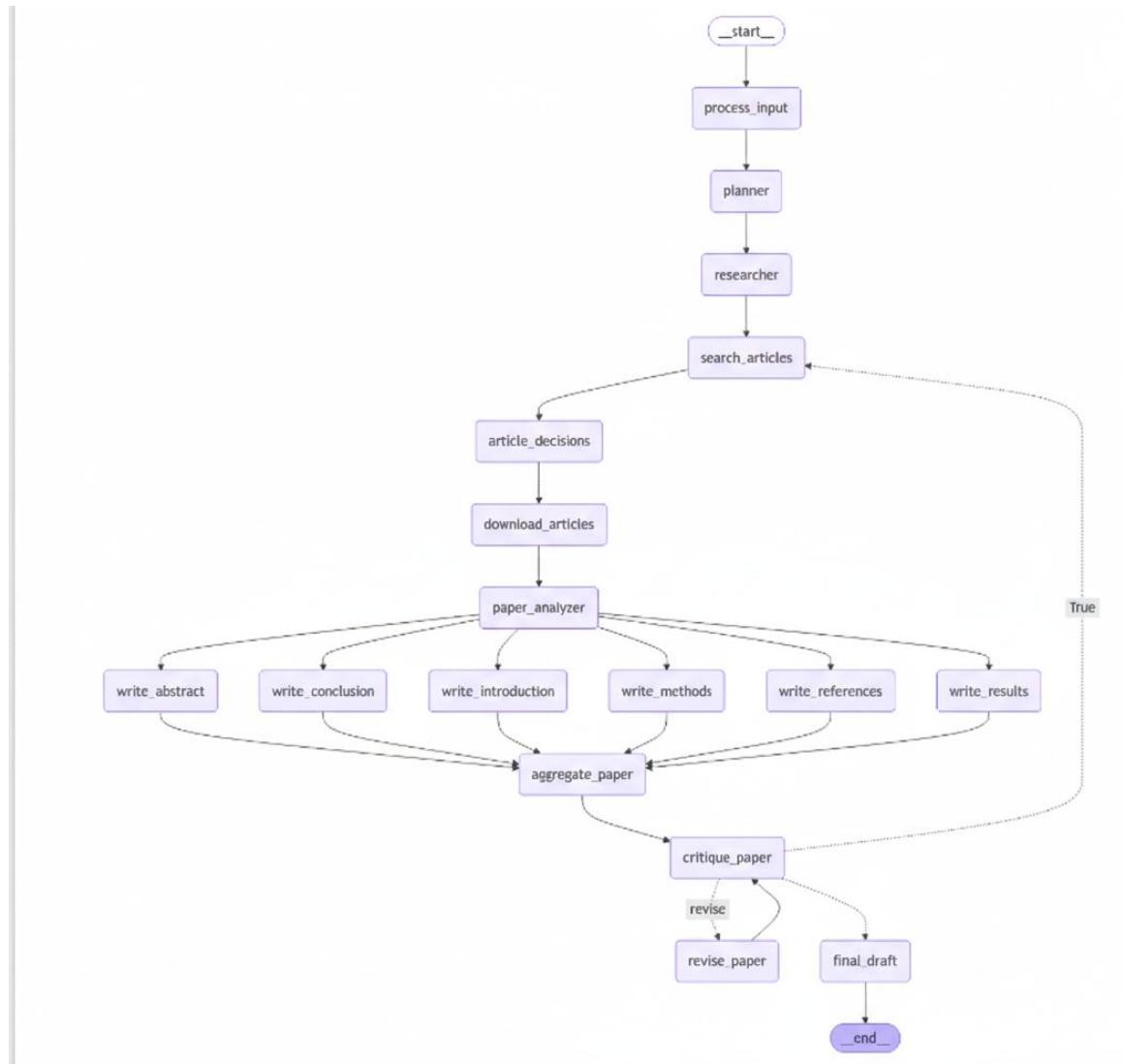
Analysis Module: Compares findings, highlights key themes, and identifies overlaps among papers.

Draft Generation Module: Generates structured sections including Abstract, Methodology, and Results using GPT-based models.

Review Module: Conducts revision checks, adds references, and prepares the final document for submission.

4. System Design / Architecture

The architecture is graph-based, enabling stateful control over each stage of the review pipeline. It ensures reliable progression from search → extraction → analysis → synthesis → revision.



5. Week-wise Module Implementation and High-Level Requirements

Milestone 1: Week 1–2

- Set up environment and install required dependencies.
- Implement automated paper search via Semantic Scholar API.
- Enable selection and download of research PDFs based on topic input.
- Prepare dataset of selected papers for analysis.

Milestone 2: Week 3–4

- Implement text extraction module for parsing downloaded PDFs.
- Extract section-wise text data and store it for analysis.
- Develop key-finding extraction logic and cross-paper comparison module.
- Validate correctness and completeness of extracted textual data.

Milestone 3: Week 5–6

- Generate automated section drafts using GPT-based models.
- Implement structured writing for Abstract, Methods, and Results.
- Integrate synthesis of findings from multiple papers.
- Format references according to APA style.

Milestone 4: Week 7–8

- Implement review and refinement cycle for generated content.
- Add revision suggestions and quality evaluation module.
- Perform Final UI Integration: Design and implement the polished Gradio **interface** to clearly present all generated sections (Abstract, Methods comparison, Results synthesis) and APA references.
- Add UI controls (e.g., 'Critique/Revise' button) to allow user-triggered re-runs of the revision cycle.
- Prepare final report combining all generated sections.
- Conduct final testing and prepare documentation and presentation deliverables.

6. Technology Stack

Programming Language: Python 3.x

Libraries / Frameworks:

- Gradio
- Grandalf
- Hugging Face Hub
- LangChain, LangGraph, LangSmith
- OpenAI API
- Pydantic, Psycopg, Tiktoken
- PyMuPDF4LLM