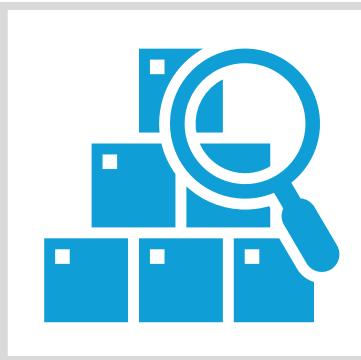




PaperPal

Introduction

PaperPal is a multi-agent AI assistant built to streamline academic and technical research workflows. Given a topic, it autonomously searches, summarizes, critiques, and simplifies recent literature — delivering insights that are both deep and accessible. It's designed to help users, from researchers to non-experts, stay informed without being overwhelmed.



Agents

#	Agent	Agent Name	Agent Goal
1		Orchestrator Agent	It coordinates the workflow by assigning the task to the right agent and managing the flow of information between them.
2		Scholar Agent	It is expert in retrieving relevant academic papers and its metadata which would be helpful in the next stages of the pipeline.
3		Market Agent	It finds the market impact of the research area / topic / technology mentioned in the user query by looking at the trending news and products / services.
4		Summarizer Agent	It summarizes all the content collected by Scholar and Market Agent so that there won't be any redundancy in the data.
5		Critic Agent	The Critic Agent evaluates research outputs by identifying gaps, biases, and limitations, ensuring the analysis is balanced and credible.
6		Report Generation Agent	The Report Generation Agent compiles insights from all agents into a well-structured, reader-friendly report with clear sections and subtopics.
7		RAG Agent	The RAG Agent leverages the summaries as a knowledge base and uses retrieval-augmented generation to answer user questions with accurate, context-aware responses.

Agents



Orchestrator Agent: Tasks

1. Directly interacts with the user
2. Identify the user query and decides which agent to call



Scholar Agent: Tasks

1. Breaks / rephrases the user query to improve the search in scholar data
2. It has a tool, Semantic Scholar API, to search and return the title, citations, publication year and abstract of relevant research papers



Market Agent: Tasks

1. It breaks / rephrases the user query to improve the search in market data
2. It has a tool, Perplexity API, to search and return the latest news, articles, trending products, services, etc., matching the user query

Agents



Summarizer Agent: Tasks

1. It gets data from Scholar Agent and Market Agent and summarizes it to eliminate redundancy



Critic Agent: Tasks

1. This agent will get the summary from Summarizer Agent and finds gaps, limitations, biases
2. Share these critics with the user which could help the user to identify a research problem



Report Generation Agent: Tasks

1. It takes summary from Summarizer Agent and critics from Critic Agent
2. Builds a well structured (with topics and sub-topics) readable report from the collected data

Agents



RAG Agent: Tasks

1. It gets summary from the Summarizer Agent and critics from the Critics Agent
2. Builds a RAG pipeline to answer any questions that user may ask

Tools: Semantic Scholar API

Semantic Scholar

- Semantic Scholar provides free, AI-driven search and discovery tools, and open resources for the global research community.
- Semantic Scholar index over 200 million academic papers sourced from [publisher partners](#), data providers, and web crawls.



Semantic Scholar

#	API	Description
1	https://api.semanticscholar.org/graph/v1/paper/search	Lets user search relevant papers for their query
2	https://api.semanticscholar.org/graph/v1/paper/{paper_id}	Lets user find detail of a particular paper with paper ID {paper_id}

Some more APIs. Please check [here](#).

API Usage

- Most Semantic Scholar endpoints are available to the public without authentication (means no API key is required), but they are limited to 1,000 requests per second shared across all unauthenticated users. At any particular moment, only first 1,000 unauthenticated requests would be served and others would be refused by the server.
- Thus, it is advised to acquire an API key which is free of cost but with a restriction of 1 request per second. All requests after the first request in a particular second of time would be refused by the server.

NOTE

1. Since it is a POC, thus only API #1 is used

Tools: Perplexity API



Perplexity AI

- Perplexity AI is an "answer engine" that provides clear, conversational answers to user queries by searching the web in real-time and summarizing the information it finds.
- Unlike a traditional search engine that returns a list of links, Perplexity uses large language models (LLMs) to provide a synthesized response with clear, inline citations to the original sources.

API

- Perplexity API lets user submit a search query and get results from all the Perplexity supported sources.
- The API is paid, and the charges depend on the number tokens and model used. Following are the per token pricing of different Perplexity models along with their description [[Perplexity API Pricing](#)].

#	Model	I/P	O/P	Citation	Search Queries	Reasoning Tokens	Best For
1	Sonar	1	1	-	-	-	Quick facts, news updates, simple Q&A
2	Sonar Pro	3	15	-	-	-	Complex queries, competitive analysis, deep research
3	Sonar Reasoning	1	5	-	-	-	Logic puzzles, math problems, transparent reasoning
4	Sonar Reasoning Pro	2	8	-	-	-	Complex problem-solving, research analysis, planning
5	Sonar Deep Research	2	8	2	5	3	Academic research, market analysis & reports

NOTE

1. Above prices are in US Dollars per million tokens
2. Since it is a POC, thus only Sonar is used

Large Language Model (LLM)

- OpenAI LLM is used in this project.
- The pricing of OpenAI API depends on the number input / output tokens and the model used.
- Below are the [pricing details](#) OpenAI API.



#	Model Name	I/P Token	Cached Token	O/P Token	Description
1	GPT – 5	1.250	0.125	10	The best model for coding and agentic tasks
2	GPT – 5 mini	0.250	0.025	2	A faster, cheaper version of GPT-5 for well-defined tasks
3	GPT – 5 nano	0.050	0.005	0.4	The fastest, cheapest version of GPT-5 for summarization and classification tasks

NOTE: Above prices are in US Dollars per million tokens

<Template>