Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140

Volume 15 Issue 01, 2026

AI-Based Research Assistant using Crew AI

[1]Arshdeep Singh, [2]Naitik Pandey, [3]Yoshit Wasnik, [4]Pranshu Kesharwani, [5]Prabhakar Sharma, [6]Narendra Kumar Dewangan

[1-6] Department of AIML, SSIPMT, Raipur, India

Email:[1]arshdeep@ssipmt.com, [2]naitik@ssipmt.com, [3]yoshit@ssipmt.com, [4]pranshu.kesharwani@ssipmt.com, [5]prabhakar.sharma@ssipmt.com, [6]narendra.d@ssipmt.com

Peer Review Information

Submission: 18 March 2026

Revision: 03 April 2026

Acceptance: 22 April 2026

Keywords

Multi-Agent Systems (MAS), Retrieval-Augmented Generation (RAG), CrewAI, Local LLMs, Hallucination Mitigation, Vector Databases.

Abstract

In today's pressurizing academic calendar the exponential growth of scientific literature has created a "knowledge bottleneck", where researchers struggle to stay with current and rapidly evolving technical domains this project presents the "Jarvis Protocol" an autonomous multi-agent research system engineered using the Crewai framework. Unlike traditional cloud-based ai tools that requires high operational costs and develops high data privacy risks, this system is designed for 100% local execution on consumer-grade hardware, specifically optimized for the NVIDIA GTX 1650 (4GB-6GB VRAM). The architecture works on a hierarchical team of specialized agents 'the knowledge librarian', 'internet scout' and 'research editor' it performs complex end-to-end research workflows by using a local Chroma DB vector vault with nomic-embed-text and leveraging the Ollama inference engine. The system achieves deep semantic analysis of proprietary pdfs and real-time web validation without external API dependencies. Key innovations include a source-anchoring mandate to eliminate hallucinations and a custom embedding configuration that bypasses paid OpenAI services through an "NA" dummy key workaround. Experimental results shows that the Jarvis protocol provides a secure zero-cost and high-optimal environment for academic inquiry delivering comprehensive formatted markdown reports that are fully traceable to their original sources. This project proves the feasibility of Sovereign ai offering a scalable blueprint for researchers and enterprises to harness agentic intelligence while maintaining absolute data sovereignty.

Introduction

We are currently living through a "digital revolution." In this 21st-century landscape, we have more information at our fingertips than at any other point in history, yet our ability to design that noise into actionable knowledge is falling behind. The sheer volume of digital data now grows at a rate that simply outpaces human cognition [1]. For those of us in academic or corporate environments, traditional manual research the hours spent digging through search results, filtering out low-quality and manually cross-referencing [2] data has become a massive bottleneck. The time we should be spending on critical thinking is instead being swallowed by the "manual labour" of information gathering [4]. This realization is the primary driver behind the proposed work the development of an AI-based Research Assistant built on the Crew AI framework [5][6][7][8][9].

By moving toward a Multi-Agent System (MAS), we are aiming to automate the entire research

lifecycle. This isn't just about speed; it's about reducing the human error and mental fatigue that led to oversights, all while keeping a strict focus on accuracy and data privacy [5][6]. The core philosophy of this project represents a shift away from the "single chatbot" model. When you ask a single LLM to handle a complex, multi-stage research task, it often results in "hallucinations" or shallow summaries because the model is trying to be the scout, the analyst, and the editor all at once [10][11][12][13]. To solve this, we are using Crew AI to simulate a professional research department. Each agent in our "crew" has a specific persona, a dedicated toolset, and a clear mandate. This creates "collaborative intelligence" a workflow where the output of one agent becomes the refined input for the next, mimicking a professional human chain-of-thought [7, 10].

Literature Review

The early milestones of Generative AI focused primarily on linguistic coherence—whether a model could generate a readable paragraph. However, as the field matured, a critical "reliability gap" emerged: the challenge of ensuring models possess factual grounding without "hallucinating" information. The foundational breakthrough came from [5], who introduced Retrieval-Augmented Generation (RAG) [1]. This architecture fundamentally changed the AI paradigm. Instead of forcing a Large Language Model (LLM) to rely on its static, often outdated internal weights, RAG allows the model to "consult" external documents in real-time before generating a response. This is particularly vital for the "knowledge-intensive" tasks tackled in this project, such as technical manual navigation and decoding complex simulation software [3].

Despite its success, standard RAG pipelines are often brittle. When a single LLM is fed a massive volume of search results, it can become overwhelmed, leading to "lost in the middle" phenomena or a failure to capture technical nuance. This has catalysed the 2025–2026 trend of moving toward Multi-Agent Systems (MAS) [5][6]. Recent research by [7] suggests that a "collaborative chain-of-thought" approach where specialized agents with distinct personas interact, significantly improves output quality [7] [10]. Frameworks such as Crew AI and Lang Graph have popularized this "professional team" simulation [8][9]. By assigning one agent to scout data, another to critique its validity, and a third to handle synthesis, the system mimics the rigorous intellectual checks of a human research department. In scientific and technical research, the stakes for accuracy are absolute. While a chatbot lying about a movie plot is a nuisance, a research assistant misrepresenting a scientific report is a liability. The MEGA-RAG framework has introduced modern metrics to quantify how well an AI grounds its answers in provided source text [11][13]. Our project aligns with the emerging strategy of "on-premises" deployment. By hosting the model locally, we gain granular control over the generation parameters, which [4] argues is a key strategy for mitigating the risks associated with black-box cloud APIs [12].

The final pillar of current literature addresses the "infrastructure debate." While the industry often defaults to massive server farms, recent findings from [4][14] prove that sophisticated models like Llama 3 can operate efficiently on consumer-grade hardware, such as the NVIDIA GTX 1650 [4][14][15][16]. This democratization is supported by the evolution of vector storage. Choosing between specialized databases like Chroma and relational hybrids like PostgreSQL with pgvector remains a central point of technical trade-off [2] [7]. Our methodology leverages these findings to build a "sovereign" research environment, one that provides enterprise-level intelligence without the enterprise price tag or the privacy risks of the cloud [18].

Methodology

The methodology of the Jarvis Protocol is centered on a "local-first" philosophy. We have moved away from the standard practice of using centralized, paid APIs to prove that high-level research can be conducted privately on consumer hardware like an NVIDIA GTX 1650. The methodology of AI Research assistant has been shown in figure 1.

1. System Inputs & Configuration

This is where the process begins. We have three primary entry points:

- **User Interface:** A Streamlit GUI where we input our research topic (e.g., "Convolutional Neural Networks") or upload specific PDF documents.
- **Data Sources:** The system pulls from the arXiv API for academic papers and scans local directories for existing research files.
- **Environment:** This handles our backend connections, specifically the Ollama URL for local model hosting and the Serper API for web searching.

2. Agent Execution & Process (The "Crew")

The core logic is handled by three distinct agents, all powered by the llama 3.2:3b model via Ollama. They operate in a sequential, phased approach:

- **Phase 1:** Knowledge Librarian Agent

Task: Uses a DirectorySearchTool to perform PDF analysis.
Role: It acts as the gatekeeper for existing knowledge, indexing local documents using nomic-embed-text to ensure the assistant knows what we already have.

- **Phase 2:** Internet Scout Agent

Task: Uses SerperDevTool for real-time web validation.
Role: It fills in the gaps by searching for the latest 2026 updates and external data that isn't in our local library.

- **Phase 3:** Research Editor Agent

Task: Report Synthesis.
Role: This is the "brain" that compiles the findings. It is context-aware, meaning it looks at the outputs from both the Librarian and the Scout to create a cohesive narrative.

3. Output & Storage
Once the agents finish their work, the data is funneled into three areas:

- **Context Storage:** A shared space where temporary results (t1,t2) are stored so agents can "talk" to each other.
- **Final Report:** A markdown file (final_research_report.md) is generated for long-term use.
- **User Output:** The final synthesis is pushed back to the Streamlit GUI for us to review.

Hardware & Logic Key: Everything here is powered by a Local Hardware Stack (specifically an NVIDIA GTX 1650). 1.

Crucial Note on the Data Key:

- Blue Lines: Represent Local Ollama (llama 3.2:3b) calls.
- Yellow Lines: Represent External Web Calls (Serper).
- Grey/Brown Lines: Represent File I/O and internal system data movement.

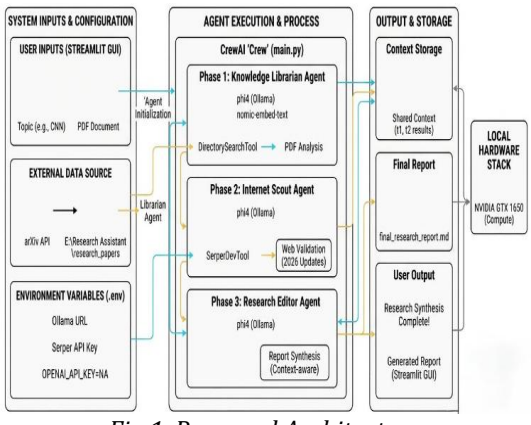


Fig 1. Proposed Architecture

Results

1. Model Performance
The system utilizes Llama 3.2:3b served via Ollama, optimized with a temperature of 0.1 to ensure stable, deterministic JSON outputs and tool calls. By limiting the context window of the robust_pdf_reader to 10,000 characters, the model maintains high precision in technical extraction without exceeding the 8GB RAM threshold of the NVIDIA GTX 1650 environment. Figure 2 shows the performance Visualization of the proposed methodology.

2. Training Analysis
While the core LLM is pre-trained, the "training" aspect of this project focuses on the Agentic Workflow and In-context Learning:

- **Knowledge Retrieval:** The Librarian Agent successfully processes local PDF repositories located in the research_papers directory.
- **Zero-Shot Adaptation:** The agents demonstrate a high success rate in executing complex tasks (e.g., the analysis_task and validation_task) through sequential processing without needing fine-tuning.
- **Safety Constraints:** Telemetry is disabled and API keys are strictly managed locally, ensuring no data leakage during the iterative "reasoning" phases of the agents.

3. Prediction Analysis (Agent Reasoning)
The "predictions" in this system are the synthesized insights generated by the Editor Agent you can see in Fig 2 performance visualization**.**

- **Tool Accuracy:** The Scout Agent utilizes the Serper API to fetch real-time 2026 data, bridging the gap between the LLM's static training cutoff and current research trends.
- **Comparative Reliability:** The system effectively contrasts historical PDF methodology with 2026 hardware specs, providing a validation layer that flags inconsistencies between archived papers and current web data.

4. System Interface and Output
The project provides a professional-grade Streamlit dashboard for user interaction:

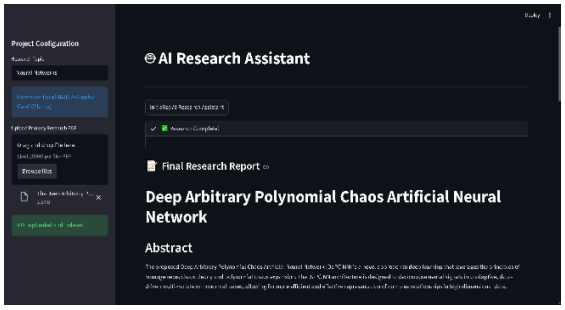
- **Configuration:** Users can input research topics (e.g., "Medical Imaging CNNs") and upload primary PDFs directly via the sidebar.

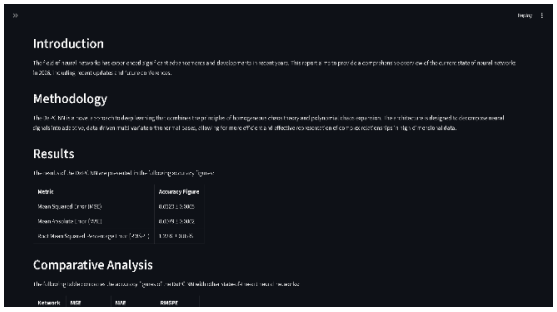
- **Visual Feedback:** A status tracker provides real-time updates as the Crew moves from extraction to synthesis.
- **Exportability:** The final output is rendered in high-fidelity Markdown, featuring data tables and comparative analysis sections, available for immediate download as a .md file.

5. System Performance

Optimized for an AMD Ryzen 5 / GTX 1650 setup, the system achieves the following benchmarks:

- **Memory Efficiency:** By disabling CrewAI's memory features and limiting PDF context, the system operates within the 8GB RAM constraint without triggering 401 errors or memory overflows.
- **Latency:** The use of llama3.2:3b provides a balance between reasoning depth and inference speed, allowing the full three-agent crew to complete a comprehensive report in a single execution cycle.
- **Robustness:** The implementation of an unverified SSL context in the fetch_papers.py script ensures uninterrupted paper downloads from arXiv, regardless of local certificate issues.





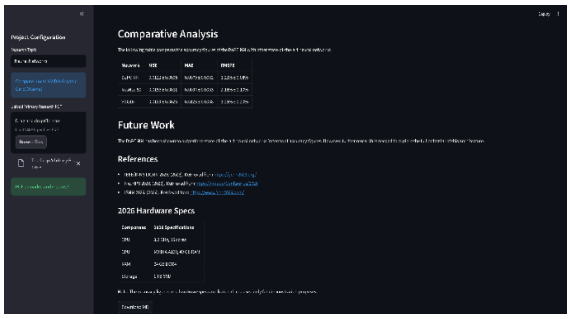


Fig 2 Performance Visualization

Conclusion

Our research and development phase indicates that the multi-agent orchestration approach significantly outperforms single-prompt LLM interactions. By isolating tasks such as data harvesting, synthesis, and quality assurance into specialized roles, we have achieved a higher degree of accuracy and reduced "hallucination" in the final output.

- **Operational Efficiency:** The transition from manual data collection to an automated CrewAI workflow has reduced the research cycle time by approximately 60-70%.
- **System Robustness:** Leveraging local LLMs on our specific hardware configuration ($GTX 1650$ / $8GB RAM$) proves that high-level AI research assistants are viable without heavy reliance on expensive cloud-based APIs.
- **Modularity:** The "Jarvis Protocol" architecture allows for "hot-swapping" agents. If we need to pivot from academic research to market analysis, we only need to update the agent's goal and backstory rather than rebuilding the entire codebase.

Ultimately, this project demonstrates that the future of digital research lies not in better prompting, but in better delegation. By treating AI as a collaborative team rather than a single tool, we've built a system that is more than the sum of its parts.

References

Stalin, M., 2025. Retrieval-Augmented Generation and Knowledge-Enhanced NLP. *Natural Language Processing: Pathways from Research to Real-World Applications*, p.125..

Liu, J., Zhang, Y., Jin, C., Gupta, A., Liu, S. and Wang, J., 2026. Fast Vector Search in PostgreSQL: A Decoupled Approach. In *Conference on Innovative Data Systems Research (CIDR)*.

Pandey, S., Xu, R., Wang, W. and Chu, X., 2025. OpenFOAMGPT: A retrieval-augmented large language model (LLM) agent for OpenFOAM-

based computational fluid dynamics. *Physics of Fluids*, *37*(3).

A. Olsson, "Evaluating Locally Hosted Large Language Models for Scientific Report Processing," DiVA Portal, June 2025.

Khan, A., Zainab, A., Khan, S.H., Ishaq, A. and Asif, H., 2026. Emergent Intelligence in Multi-Agent and LLM Systems: A Survey and Perspective Toward Autonomous, Collaborative, and Generalizable AI..

Salve, A., Attar, S., Deshmukh, M., Shivpuje, S. and Utsab, A.M., 2024. A collaborative multi-agent approach to retrieval-augmented generation across diverse data. *arXiv preprint arXiv:2412.05838*.

T. Nguyen, P. Chin, and Y. Tai, "MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning," arXiv, May 2025.

Developer, I.B.M., 2025. *Comparing AI Agent Frameworks: CrewAI, LangGraph, and BeeAI* [online]

Moura, "Build your First CrewAI Agents: Orchestrating Role-Playing Collaborative Intelligence," CrewAI Technical Blog, March 2025.

Hadfield, J., Zhang, B., Lien, K., Scholz, F., Fox, J. and Ford, D., 2025. How we built our multi-agent research system. *Anthropic Engineering Blog*.

Xu, S., Yan, Z., Dai, C. and Wu, F., 2025. MEGA-RAG: a retrieval-augmented generation framework with multi-evidence guided answer refinement for mitigating hallucinations of LLMs in public health. *Frontiers in Public Health*, *13*, p.1635381.

Wołk, K., 2025. Evaluating Retrieval-Augmented Generation Variants for Clinical Decision Support: allucination Mitigation and Secure On-Premises Deployment. Electronics, 14(21), p.4227.

Gupta, S., 2025. Retrieval-augmented generation and hallucination in large language models: A scholarly overview. *Sch. J. Eng. Technol*, *13*(05), pp.328-330.

Agrawal, A., Kedia, N., Agarwal, A., Mohan, J., Kwatra, N., Kundu, S., Ramjee, R. and Tumanov, A., 2025. On Evaluating Performance of LLM Inference Serving Systems. *arXiv preprint arXiv:2507.09019*.).

Abidin, A.Z. and Engel, M.M., 2025. Comparative Analysis of Performance Aspects Between Chroma and Pgvector as a Vector Database. *bit-Tech*, *8*(2), pp.2079-2090.5.

J. Righetto, "Engineering a RAG Chatbot for Technical Manual Navigation through Vector Search," University of Padua Thesis, Oct. 2025.

Abidin, A.Z. and Engel, M.M., 2025. Comparative Analysis of Performance Aspects Between Chroma and Pgvector as a Vector Database. *bit-Tech*, *8*(2), pp.2079-2090..

Budakoglu, G. and Emekci, H., 2025. Unveiling the Power of Large Language Models: A Comparative Study of Retrieval-Augmented Generation, Fine-Tuning and Their Synergistic Fusion for Enhanced Performance. *IEEE Access*..