2026 Volume: 02 Issue: 05 www.ijrpa.com ISSN 2456-9995 Review Article

International Journal Research Publication Analysis

Page: 01-07

DESIGN AND DEVELOPMENT OF AN AI-POWERED MULTI-AGENT DOCUMENT ANALYSIS AND REPORTING SYSTEM USING RAG AND LANGGRAPH

***Sujal Thakur**

Department of Computer Science & Artificial Intelligence.

Article Received: 08 April 2026
Article Revised: 28 April 2026
Published on: 18 May 2026

***Corresponding Author: Sujal Thakur**
Department of Computer Science & Artificial Intelligence.

DOI: https://doi-doi.org/101555/ijrpa.3050

ABSTRACT

This paper presents the design and development of an AI-powered multi-agent system for intelligent document analysis and automated report generation. The proposed architecture integrates Retrieval-Augmented Generation (RAG) with LangGraph-based agent orchestration to enable dynamic, intent-driven workflows. Specialized agents handle document retrieval, contextual analysis, chart generation, and structured report synthesis. An intent classification module routes user queries to the appropriate agent pipeline, while a validation agent mitigates hallucination through RAG grounding and fact-verification. Experimental evaluation demonstrates that the proposed system achieves 93.7% retrieval accuracy, reduces hallucination rate to 4.2%, and generates comprehensive reports in under 8 seconds—outperforming baseline single-agent and standalone RAG approaches. The modular architecture ensures extensibility across diverse document domains.

KEYWORDS: *Multi-Agent Systems, Retrieval-Augmented Generation (RAG), LangGraph, Document Analysis, Natural Language Processing, Generative AI, Hallucination Prevention, Report Automation*

I. INTRODUCTION

The exponential growth of unstructured data in organizational and research environments necessitates intelligent systems capable of automated document comprehension and insight extraction. Traditional document processing pipelines rely on monolithic architectures that lack flexibility, interpretability, and adaptability to diverse query types.

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding; however, their tendency to generate plausible yet factually incorrect outputs—commonly termed hallucination—remains a critical challenge in production deployments. Retrieval-Augmented Generation (RAG) addresses this limitation by grounding model responses in retrieved document context, thereby improving factual fidelity.

This work proposes a novel multi-agent architecture that combines RAG-based document retrieval with LangGraph-orchestrated agent workflows. The system employs an intent-based routing mechanism that dynamically dispatches queries to specialized agents: retrieval, analysis, chart generation, report synthesis, and validation. This modular design enables the system to handle complex, multi-step document analysis tasks that exceed the capabilities of single-agent or retrieval-only systems.

The primary contributions of this paper are: (1) a modular multi-agent framework integrating RAG and LangGraph; (2) an intent classification module for dynamic agent routing; (3) a hallucination prevention pipeline using RAG grounding; (4) automated chart generation from document-extracted data; and (5) comprehensive empirical evaluation demonstrating significant performance improvements over baseline approaches.

II. RELATED WORK

A. Retrieval-Augmented Generation

Lewis et al. [1] introduced RAG as a paradigm combining parametric knowledge from pre-trained language models with non-parametric retrieval from external knowledge stores. Subsequent work by Gao et al. [2] extended RAG with advanced retrieval strategies including iterative and adaptive retrieval, demonstrating improved performance on knowledge-intensive NLP tasks.

B. Multi-Agent LLM Systems

AutoGPT [3] and BabyAGI [4] pioneered autonomous multi-agent architectures where LLMs decompose and execute complex tasks through iterative planning and tool use. LangGraph [5], built atop LangChain, provides a graph-based state machine formalism for defining cyclical multi-agent workflows, enabling more controllable and debuggable agent pipelines.

C. Hallucination Mitigation

Factual inconsistency in LLM outputs has been addressed through diverse strategies, including chain-of-thought prompting [6], self-consistency decoding [7], and knowledge-grounded generation [8]. Our system builds upon these insights by implementing a dedicated validation agent that cross-references generated content against retrieved document chunks.

III. SYSTEM ARCHITECTURE

The proposed system comprises six specialized agents orchestrated through a LangGraph state machine, as illustrated in Fig. 1. The architecture adopts a directed acyclic graph (DAG) topology with conditional edges that enable dynamic routing based on intent classification outputs.

[Fig. 1: System Architecture — User Query → Intent Router → {RAG Agent | Analysis Agent | Chart Agent | Report Agent} → Validation Agent → Output]

Fig. 1: High-level architecture of the proposed AI-powered multi-agent document analysis system.

A. Intent Classification Module

The intent router employs a fine-tuned prompt-based classifier that categorizes incoming queries into five operational intents: (i) document retrieval, (ii) analytical summarization, (iii) data visualization, (iv) report generation, and (v) factual verification. Classification is performed using a zero-shot chain-of-thought prompting strategy with GPT-4, achieving 91.5% classification accuracy on the evaluation benchmark.

B. RAG Retrieval Agent

Documents are preprocessed through a chunking pipeline that segments text into 512-token overlapping windows with a 64-token stride. Chunks are encoded using OpenAI text-embedding-ada-002 and indexed in a FAISS vector store. At query time, the retrieval agent performs cosine similarity search to retrieve the top-k (k=5) most relevant chunks, which are prepended to the LLM context window as grounding evidence.

C. Analysis and Synthesis Agents

The analysis agent applies extractive and abstractive summarization over retrieved chunks using GPT-4 with structured output prompts. The report synthesis agent aggregates outputs

from multiple agents into a coherent structured document, applying IEEE-style section templates for academic report generation.

D. Chart Generation Agent

Numerical data extracted from documents is passed to the chart agent, which employs Pandas for data manipulation and Matplotlib for visualization. The agent supports bar charts, line graphs, scatter plots, and pie charts, with automatic axis labeling and title generation derived from document context.

IV. AGENT ROLES AND TECHNOLOGIES

TABLE I: Multi-Agent System — Agent Roles and Technologies.

Agent	Responsibility	Technology Used
Intent Router	Classifies user queries and routes to appropriate agent	LangGraph, Prompt Engineering
RAG Retrieval Agent	Retrieves semantically relevant document chunks	FAISS, OpenAI Embeddings, LangChain
Analysis Agent	Performs NLP-based document analysis and summarization	GPT-4, LangChain
Chart Agent	Generates dynamic visualizations from extracted data	Matplotlib, Pandas
Report Agent	Synthesizes findings into structured reports	GPT-4, FPDF/DOCX
Validation Agent	Verifies output accuracy and prevents hallucination	Fact-checking prompts, RAG grounding

V. IMPLEMENTATION

The system is implemented in Python 3.11 using the LangChain v0.2 and LangGraph v0.1 frameworks. The LLM backbone is GPT-4-turbo-preview accessed via the OpenAI API. Vector storage and retrieval are implemented using FAISS (Facebook AI Similarity Search) with L2-normalized cosine distance. The web interface is developed using Streamlit, enabling interactive document upload and real-time agent execution visualization.

The LangGraph workflow is defined as a StateGraph with typed state annotations, ensuring type safety across agent boundaries. Each agent node receives the full conversation state and returns a partial state update, enabling efficient incremental processing. Conditional edges implement the intent-routing logic, evaluating the classifier output stored in the shared state object.

Document ingestion supports PDF, DOCX, TXT, and CSV formats. The preprocessing pipeline applies OCR (via Tesseract) for scanned PDFs, followed by text normalization, sentence segmentation, and metadata extraction (title, author, date, page numbers). Extracted metadata is stored alongside embeddings to enable metadata-filtered retrieval.

VI. EXPERIMENTAL EVALUATION

A. Dataset and Experimental Setup

Evaluation was conducted on a curated benchmark of 150 documents spanning research papers, technical reports, and financial documents, with 500 manually annotated query-answer pairs. System performance was assessed on five metrics: retrieval accuracy, hallucination rate, report generation latency, intent classification accuracy, and user satisfaction (5-point Likert scale, n=30 evaluators).

TABLE II: Performance Comparison with Baseline Systems.

Metric	RAG-Only	LangGraph-Only	Proposed System	Human Baseline
Retrieval Accuracy (%)	81.2	74.5	93.7	96.1
Hallucination Rate (%)	14.3	18.6	4.2	1.8
Report Generation (sec)	12.4	9.8	7.3	N/A
Intent Classification (%)	N/A	79.3	91.5	97.2
User Satisfaction (/5)	3.6	3.4	4.5	4.8

Table II: Quantitative comparison of proposed system vs. baselines across evaluation metrics.

B. Analysis

The proposed system achieves a 15.4% improvement in retrieval accuracy over standalone RAG (93.7% vs. 81.2%), attributable to the intent-conditioned query reformulation performed by the intent router prior to retrieval. The hallucination rate is reduced from 14.3% to 4.2%, representing a 70.6% relative reduction, demonstrating the efficacy of the validation agent and RAG grounding strategy.

Report generation latency of 7.3 seconds represents a 41% improvement over the RAG-only baseline (12.4s), achieved through parallel agent execution enabled by the LangGraph asynchronous node execution model. User satisfaction scores of 4.5/5 indicate strong acceptance of the generated report quality.

VII. DISCUSSION

The modular multi-agent architecture demonstrates significant advantages over monolithic approaches in terms of maintainability, extensibility, and performance. The separation of concerns between retrieval, analysis, visualization, and synthesis enables independent optimization of each component without disrupting the overall pipeline.

Limitations of the current system include dependency on proprietary LLM APIs (OpenAI GPT-4), which introduces cost and latency constraints for large-scale deployment. Future work will investigate the integration of open-source LLMs (Llama 3, Mistral) as cost-effective alternatives. Additionally, the current vector store implementation uses a flat FAISS index; hierarchical navigable small world (HNSW) indexing will be explored for improved scalability.

VIII. CONCLUSION

This paper presented a novel AI-powered multi-agent document analysis and reporting system integrating RAG and LangGraph. The system addresses key limitations of existing approaches through intent-based dynamic routing, specialized agent decomposition, and active hallucination prevention. Experimental results demonstrate state-of-the-art performance across retrieval accuracy, factual consistency, generation latency, and user satisfaction metrics. The proposed architecture provides a robust foundation for intelligent document processing applications in research, enterprise, and educational domains.

REFERENCES

1. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020.
2. Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023.
3. T. Significant Gravitas, "AutoGPT: An Autonomous GPT-4 Experiment," GitHub Repository, 2023.
4. Y. Nakajima, "BabyAGI: Task-Driven Autonomous Agent," GitHub Repository, 2023.

5. H. Chase, "LangGraph: Building Stateful, Multi-Actor Applications with LLMs," LangChain Documentation, 2024.
6. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," NeurIPS, 2022.
7. X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," ICLR, 2023.
8. S. Shuster et al., "Retrieval Augmentation Reduces Hallucination in Conversation," EMNLP Findings, 2021.