ASSESSING RESEARCH PAPERS USING MULTI-AGENTIC RAG

TEAM BLACK BOX CREW

TABLE OF CONTENTS

INTRODUCTION 01
Overview 01

METHODOLOGY 02-06
Parsing 02
Extractor Agent 02
Multi Agent Framework 02
Decision Making Agent 02
Task 1 - Architecture 03
Discussion 04
Pathway Integration 04
Conference Recommendation Agent 04
Sufficiency Judgement Agent 04
Final Decision Making Agent 05
State Graph 05
Approaches used for latency reduction 05
Key Features 05
Task 2 - Architecture 06
Discussion 06

RESULTS 07

TECHNOLOGIES USED 08
Dynamic Multi Agentic RAG 08

CONCLUSION 09

INTRODUCTION

Large Language Models (LLMs) excel at information processing but face challenges like outdated knowledge and hallucinations. Retrieval-Augmented Generation (RAG) tackles these by integrating external data sources for grounded outputs. Building on this, **Agentic RAG** equips LLMs with iterative retrieval and adaptive reasoning, enabling proactive problem-solving. This framework underpins our tasks, enhancing the assessment of research paper quality and optimizing conference selection with real-time, data-driven insights.

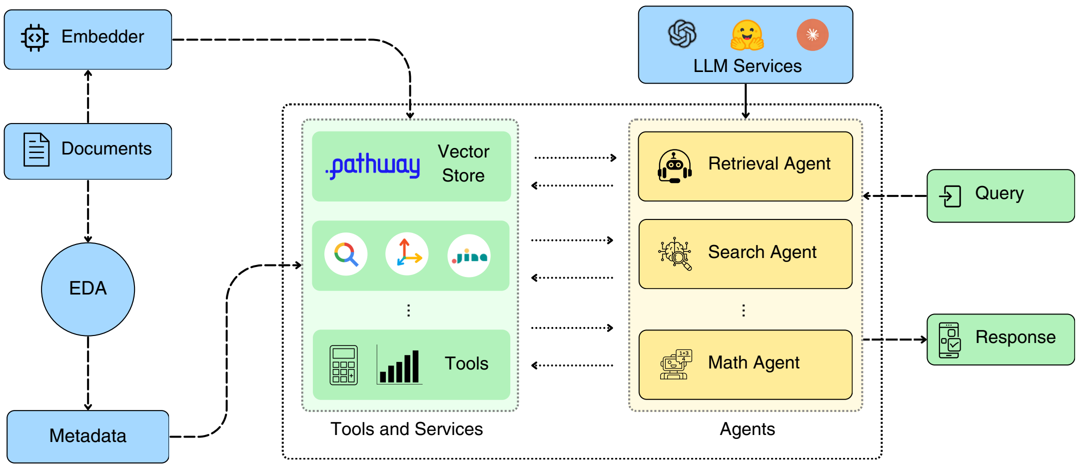


Fig 1: Multi-agent RAG system

Task 1: Research Paper Publishability Assessment

The objective of Task 1 is to develop a framework capable of evaluating the quality and publishability of research papers. With the growing volume of academic publications, this task addresses the need for an automated system to classify papers as either "Publishable" or "Non-Publishable." The framework analyzes critical aspects such as methodology, coherence, and **evidence validation** to ensure an objective evaluation. The challenge involves building a robust classification system that can accurately detect potential issues in a paper's content, making it applicable across various research domains.

Task 2: Conference Selection

Task 2 focuses on developing a system that analyzes a research paper's content and suggests the most appropriate conference for submission. The framework evaluates the paper's subject matter, methodology, and findings, comparing them against a list of prestigious conferences like CVPR, NeurIPS, EMNLP, and KDD. The system provides a recommendation along with a justification, highlighting how the paper's contributions align with the conference's themes. The framework is enhanced by integrating real-time data streaming through **Pathway's connectors**, ensuring efficient access and analysis. This task builds on the classification system from Task 1 to ensure that only "publishable" papers are considered for the conference selection process.

Our pipeline leverages a robust architecture to automate research paper evaluation and conference selection. It begins with a **LLama parser**, designed to parse research paper content into structured components. An **extractor agent** then identifies key sections such as the abstract, methodology, objectives, results, and conclusions. Following this, four specialized agents independently assess critical aspects: novelty, methodology, research objectives, and results. A decision-making agent aggregates these evaluations to deliver a final verdict.
We utilized the **LangGraph** framework for orchestrating agent interactions, and the **Groq** API for high-performance inference. The pipeline integrates state-of-the-art models like LLama and Mistral, ensuring precision and scalability in real-time, data-driven analysis.

Parsing

We utilized the LLama Parser to extract structured content from research papers with high precision. By employing a **custom parsing instruction**, we directed the parser to retrieve essential elements such as text, equations, tables, methodologies, results, discussions, and conclusions. This ensured that every critical detail from the research paper was captured comprehensively.

The parsing process was implemented using the LlamaParse library, with outputs formatted in Markdown for easy integration into our pipeline. This extracted content served as the foundation for subsequent analysis by specialized agents, streamlining the evaluation and decision-making processes.

Extractor Agent

The problem addressed in this project is the extraction and analysis of key information from research papers in markdown format to streamline understanding and categorization of their content. Researchers and professionals often encounter challenges in navigating lengthy and complex papers to identify critical elements like research topics, methodologies, results, and novelty claims. This tool leverages language models and asynchronous processing to automate the extraction of such features from structured sections of papers, improving efficiency and reducing manual effort. By caching results and using token rate-limiting, the solution is designed for scalability and can process multiple documents in parallel, making it a robust system for academic or organizational research analysis.

Multi Agent Framework

We have defined four specialized agents for evaluating research papers: MethodologyAgent, ResearchObjectiveAgent, ResultsAgent, and NoveltyClaimsAgent.

Research Objective	Novelty Agent	Methodology Agent	Results Agent
Defines clear goals, refines hypotheses, and aligns objectives with the broader research context.	Identifies and introduces new ideas, techniques, and emerging trends to disrupt the status quo.	Designs structured, repeatable, and scientifically sound methods for efficient problem-solving.	Interprets data, draws conclusions, and evaluates the success and relevance of the research outcomes.

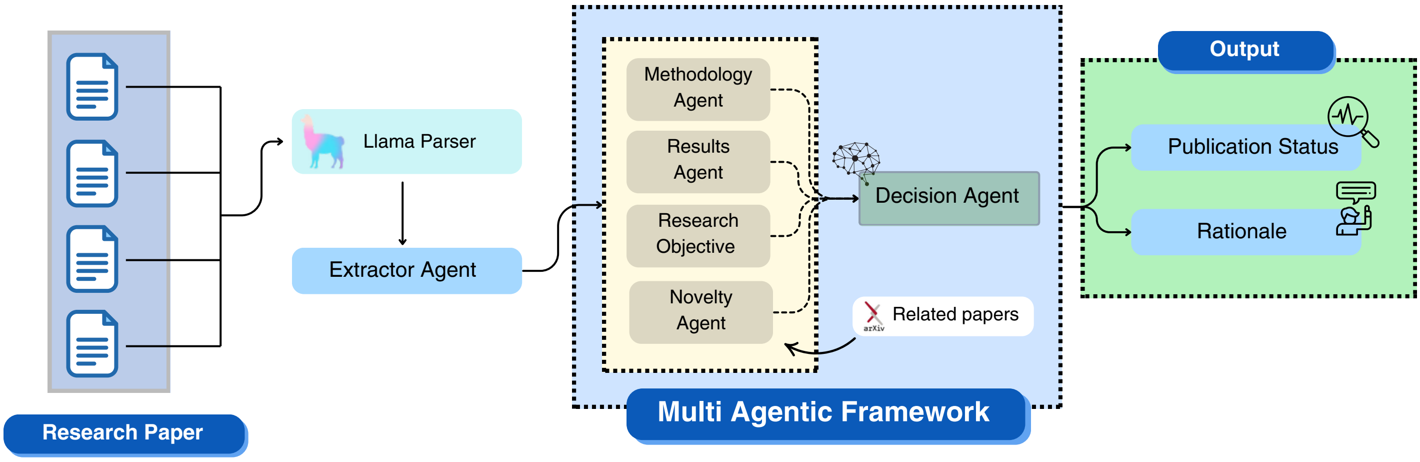
Each agent focuses on specific aspects of research evaluation, including methodology, research objectives, results, and novelty claims, respectively. Using **structured prompts** and analysis frameworks, the agents critically assess papers for strengths, weaknesses, and overall quality in their respective domains. Each agent uses predefined critical issues to identify methodological flaws, goal clarity, result validity, or innovation claims. The output is synthesized into comprehensive assessments, highlighting critical issues that may affect the **publishability** of the research. The architecture suggests the use of language models (e.g., ChatGroq) to process and evaluate text content systematically.

Decision Making Agent

This agent evaluates a paper's publishability by applying strict criteria based on analyses from the 4 specialized agents. It assesses key aspects such as methodological consistency, hypothesis clarity, statistical validity, and evidence support, **identifying critical issues** like flawed methods, weak hypotheses, and unsupported claims. Using a structured decision-making process, it classifies the paper as "Publishable" or "Non-Publishable" by evaluating argument coherence, empirical validation, and the appropriateness of methodologies. By synthesizing these analyses, the DecisionMakingAgent ensures decisions are **evidence-driven** and adhere to rigorous academic standards.

METHODOLOGY

Task-1 Architecture



Discussion

The architecture for research paper publishability assessment leverages a structured pipeline to evaluate the quality and suitability of research papers for publication. It begins with **input research papers**, which are processed through the **Llama Parser** to extract structured components like the title, abstract, methodology, results, and discussion. This parsed data is passed to the **Extractor Agent**, which identifies and isolates key elements such as the methodology, research objectives, results, and novelty. The extracted information is then analyzed within a **Multi-Agentic Framework**, composed of specialized agents: the **Methodology Agent** evaluates the rigor of the research methods, the **Results Agent** examines the significance of the findings, the **Research Objective Agent** assesses the clarity and relevance of the research goals, and the **Novelty Agent** determines the originality and contribution of the paper to its field. These agents collectively feed their evaluations into the **Decision Agent**, which synthesizes the findings and makes a final judgment about the paper's publishability. The framework provides a comprehensive output comprising the Publication Status—labeling the paper as either "Publishable" or "Non-Publishable"—and a detailed Rationale, outlining the reasons for the decision. This transparent feedback ensures accountability while addressing potential weaknesses or strengths in the paper. Designed for scalability and objectivity, the system ensures efficient evaluation across diverse academic domains, reducing subjectivity in traditional peer review processes. Additionally, its multi-agent approach ensures that critical aspects such as methodology rigor, evidence validation, coherence, and novelty are holistically analyzed, resulting in a robust and reliable classification of research papers. This pipeline aims to streamline the review process, enhance research quality, and make evaluations accessible to researchers across disciplines.

Pathway Integration

The system integrates a lightweight vector document store using Pathway's vector store as a reference model. Pathway's vector store is designed for efficient storage and retrieval of high-dimensional vectors, such as embeddings of academic papers, enabling fast similarity searches. In this implementation, a caching mechanism mimics the functionality of a vector store by storing precomputed results of similarity searches. Each research paper is **hashed** using **SHA-256** to generate a unique cache key, which ensures efficient retrieval of similar papers for a given conference. When a paper is analyzed, the system first checks the **cache** for existing results. If a match is found, the cached data is reused, avoiding redundant computations. If no match exists, the system performs a similarity search, **retrieves relevant papers**, and stores the results in the cache for future use. The integration of caching aligns with the principles of Pathway's vector store, **emphasizing efficiency** and performance in handling large-scale document retrieval tasks.
Note: We have added 5 extra papers per conference to the database to improve our results.

Conference Recommendation Agent

The Conference Recommendation Agent is the core component responsible for evaluating the alignment of a research paper with the themes of specific conferences. It leverages Large Language Models (LLMs) to analyze the paper's content, methodologies, and key topics, comparing them against the typical themes of each conference. The agent retrieves similar papers from a cached vector store to provide context and enhance the analysis. Using a structured prompt template, it generates a detailed analysis, identifying reasons why the paper should or should not belong to a particular conference. Additionally, the agent **incorporates fact-checking** to verify the **accuracy of its analysis**, ensuring reliable and well-reasoned outputs. By processing multiple conferences in parallel, the agent efficiently evaluates the **paper's suitability** across various domains, providing a comprehensive foundation for the final recommendation. This approach combines advanced LLM capabilities with domain-specific knowledge to deliver precise and insightful conference recommendations.

Tools Used

Arxiv Tool

The Arxiv tool is integrated into the system to retrieve academic papers and verify references, ensuring the accuracy and credibility of the analysis. It allows the agent to access a vast repository of research papers, enabling fact-checking and providing additional context for the paper under evaluation. This tool is particularly useful for retrieving similar papers or verifying claims made in the research paper, enhancing the robustness of the conference recommendation process.

Google Search Tool

The Google Search tool (using GoogleSearchAPIWrapper) complements the Arxiv tool by performing web searches to gather additional information or context that may not be available in academic databases. It helps the agent retrieve real-time data, such as recent developments, related publications, or conference-specific details, further enriching the analysis.

Sufficiency Judgment Agent

The Sufficiency Judgment Agent evaluates the quality and completeness of the analyses generated by the Conference Recommendation Agent. Using a separate LLM, it assesses whether the analysis is **comprehensive**, **well-reasoned, and sufficient** to make a reliable recommendation. If the analysis is deemed insufficient, the agent provides suggestions for improvement, prompting a re-evaluation. If the analysis meets the required standards, it confirms its **validity**. This step ensures that only high-quality, well-justified analyses are considered for the final decision, enhancing the overall **reliability** and **accuracy** of the conference recommendation system. The agent acts as a critical quality control mechanism in the workflow.

Final Decision-Making Agent

The Final Decision-Making Agent is responsible for synthesizing all the analyses generated by the Conference Recommendation Agents and determining the most suitable conference for the research paper. It uses a dedicated LLM to compare the analyses across different conferences, evaluating their alignment with the paper's content, methodologies, and themes. The agent identifies the conference that best matches the paper's focus and provides a detailed, **well-reasoned recommendation**. This final step ensures that the decision is based on a comprehensive evaluation of all available insights, **balancing the strengths and weaknesses** of each analysis. By aggregating and interpreting the results, the Final Decision-Making Agent delivers a clear and actionable recommendation, completing the multi-agent workflow.

State Graph

The State Graph orchestrates the **workflow of the multi-agent system**, managing the flow of data and tasks between the Conference Recommendation Agent, Sufficiency Judgment Agent, and Final Decision-Making Agent. It defines the sequence of operations, starting with the **parallel analysis of the paper** across multiple conferences, followed by the sufficiency evaluation of each analysis, and culminating in the final recommendation. The graph ensures that each step is executed in the correct order and that the results from one stage are seamlessly passed to the next. By structuring the workflow as a state graph, the system achieves modularity, scalability, and clarity, enabling efficient processing and easy extension of the pipeline. This design is critical for **handling complex**, multi-step decision-making processes in a systematic and organized manner.

Approaches used for latency reduction

Multi-Threading Approach

The system employs a multi-threading approach using ThreadPoolExecutor to process multiple conferences in parallel. Each conference analysis is handled by a separate thread, allowing simultaneous execution of tasks. This significantly reduces processing time, especially when analyzing a paper against multiple conferences.

Caching Technique

A TTLCache is implemented to store and reuse results of similarity searches for papers. By hashing each paper and caching similar papers, redundant computations are avoided. This reduced latency by 4 times, as cached results are retrieved instantly instead of performing repeated searches. The cache also ensures scalability by managing memory usage efficiently.

Key Features

Parallel Processing: Conferences are analyzed in parallel for efficiency.

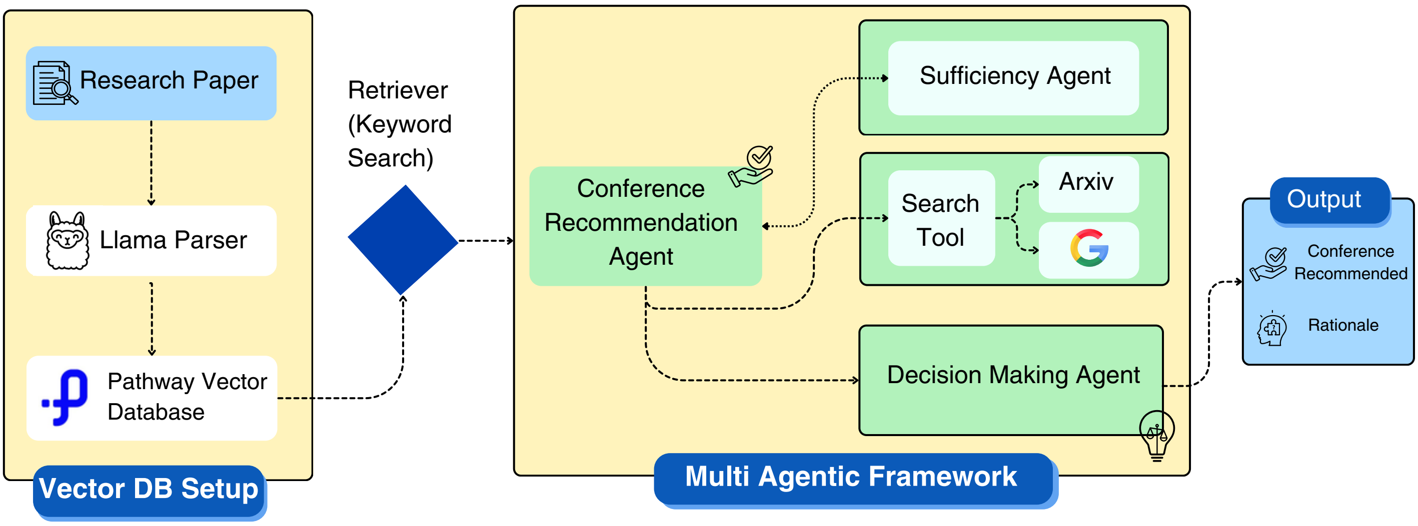
Sufficiency Judgment: Ensures high-quality analyses.

Caching: Similar papers are cached to avoid redundant computations

Fact-Checking: Analyses are verified for accuracy

Final Aggregation: Combines all analyses to make a well-reasoned recommendation

Task-2 Architecture



Discussion

The Task-2 pipeline is a multi-agent architecture designed to recommend the most suitable conference for research paper submission. The process begins with the input of a research paper, which is parsed using a **Llama Parser** to extract essential information such as the title, abstract, and introduction and conclusion . This data is then converted into high-dimensional embeddings using a Pathway Vector Database setup. These embeddings serve as the foundation for similarity-based retrieval and ensure efficient storage and analysis of the research paper's content.

The core of the system is the **Conference Recommendation Agent**, which evaluates the research paper against the themes of various conferences. Using the vector embeddings, it performs comparisons with predefined conference topics to identify potential matches. To ensure accuracy and context, the system integrates multiple tools within its framework. The Sufficiency Agent acts as a quality control mechanism, assessing whether the analysis is comprehensive and well-reasoned. It also incorporates external tools like **Arxiv** to retrieve related academic papers and **Search tools** to gather additional information about the conferences, including recent trends or updates that may influence the recommendation.

Once the **Sufficiency Agent** confirms the validity of the analysis, the **Decision-Making Agent** consolidates all insights, weighing the alignment of the research paper with the themes of each conference. This ensures a thorough evaluation across multiple parameters, ultimately determining the best-suited conference. The output includes the recommended conference along with a rationale explaining how the research aligns with the conference's focus areas.

This structured workflow is orchestrated within a multi-agent framework, ensuring seamless communication between components. By leveraging tools for analysis, verification, and decision-making, the pipeline delivers reliable, transparent, and efficient recommendations for conference submissions.

RESULTS

TASK 1

Paper No.: P099

Enhancing LSTM-based Video Narration Through Text-Derived Linguistic Insights

The decision to classify the paper as publishable is based on its significant strengths, including its **novel methodology**, unique approach to novelty claims, and **clear research objective**. The issues identified can be addressed through minor revisions, and the paper has the potential to make a **substantial impact** on the field of **video description and accessibility**.

Paper No.: P055

Examining the Initial Experiences of Researchers When Articulating Broader Impact

The decision to classify the paper as **Non-Publishable** is based on the major issues identified, including the **lack of content** in the NOVELTY_CLAIMS section, the **lack of methodological detail** in the RESULTS section, and the **narrow focus** of the RESEARCH_OBJECTIVE section. While the paper has some significant strengths, including the valuable insights provided by the RESULTS section and the strong METHODOLOGY section, the major issues need to be addressed through revisions before the paper can be considered for publication.

TASK 2

Paper No.: P001 **Leveraging Clustering Techniques for Enhanced Drone Monitoring and Position Estimation**	**Analysis and Comparison for Conferences** 1. **Thematic Alignment**: CVPR's focus on 3D vision, robotics, and object detection closely matches the paper's themes of drone monitoring, LiDAR-based 3D estimation, and tracking. 2. **Methodology Fit:** The clustering techniques used (DBSCAN, K-Means) align with CVPR's focus, unlike NeurIPS (neural systems) or EMNLP (language processing). 3. **Relevant Publications**: Similar work has been published at CVPR, ensuring the audience and reviewers are well-suited. 4. **Broader Interest**: The application of computer vision to real-world problems like drone monitoring appeals to CVPR's broad community. 5. **Other Conferences**: NeurIPS and EMNLP are unsuitable, while KDD and TMLR have limited thematic relevance compared to CVPR. CVPR is the best fit for the paper's themes, methodology, and impact.
Recommended Conference	**CVPR (Conference on Computer Vision and Pattern Recognition)**

TECHNOLOGIES USED

Dynamic Multi Agentic RAG

Description

The Dynamic Multi-Agentic Retrieval-Augmented Generation system leverages multiple agents that interact in real-time to handle diverse tasks within a research or problem-solving context. Using LangGraph, each agent is designed to fulfill specific roles, such as discovering novel techniques, structuring methodologies, or analyzing results

Embedding Generation

The system uses Sentence Transformers with the all-MiniLM-L6-v2 model to generate embeddings for research papers. This model is a lightweight, pre-trained transformer model optimized for generating high-quality sentence embeddings. It converts the text of each paper into a dense vector representation, capturing semantic meaning and enabling efficient similarity comparisons.

Problem Faced

The system initially struggled with incomplete or inaccurate analyses due to limited context and lack of external data integration. This led to unreliable conference recommendations, as the LLMs lacked access to relevant similar papers or conference-specific information.

Solution

The Dynamic Multi-Agentic RAG architecture integrated tools to retrieve external data, enriching the analysis. This ensured accurate, context-aware recommendations by combining the LLMs' reasoning capabilities with real-world information, significantly improving the system's reliability and precision.

LangGraph Integration

The system employs a dynamic multi-agentic RAG architecture, where each agent performs a specialized task and contributes to the overall decision-making process. The State Graph in LangGraph acts as the backbone of this architecture, defining the flow of data and tasks between agents. The graph orchestrates Parallel Analysis by the Conference Recommendation Agent, Sufficiency Evaluation by the Sufficiency Judgment Agent, and Final Aggregation by the Final Decision-Making Agent for a comprehensive recommendation process.

Retrieval Methods

The system computes cosine similarity between the input paper's embedding and conference paper embeddings to identify relevant papers. Results are cached using TTLCache, with papers hashed via SHA-256 for quick retrieval, reducing redundancy. Integration with Arxiv and Google Search retrieves additional context, enhancing analysis quality. This combination ensures efficient, accurate, and scalable conference recommendations.

Problem Faced

The workflow initially faced challenges in managing the flow of data between multiple agents, leading to inefficiencies and errors. The lack of a structured framework made it difficult to handle parallel processing and ensure seamless collaboration between agents.

Solution

LangGraph provided a structured state graph to orchestrate the workflow, enabling efficient data flow and parallel processing. It ensured modularity, scalability, and error handling, making the system robust and adaptable to complex multi-agent tasks.

Multi Threading

We use ThreadPoolExecutor from Python's concurrent.futures module to enable parallel processing of conference analyses. This technology allows the system to simultaneously evaluate the research paper against multiple conferences, significantly reducing processing time. By distributing tasks across multiple threads, ThreadPoolExecutor ensures efficient resource utilization and scalability, making the system capable of handling large-scale analyses without compromising performance. This parallel execution is critical for maintaining responsiveness and speed in the multi-agent workflow.

Conclusion, Challenges, Insights, and Future Directions

Conclusion

The pipeline demonstrates a sophisticated approach to conference recommendation by leveraging dynamic multi-agent collaboration. Each agent performs specialized tasks—ranging from parallel analysis and quality evaluation to final decision-making—ensuring a data-driven and systematic workflow. This integration results in an efficient and robust process, capable of delivering accurate and well-informed recommendations.

Challenges

Implementing this pipeline presented several challenges, particularly in managing API-related limitations. Ensuring seamless synchronization between agents required overcoming rate limits imposed by external APIs, which often restricted the number of requests within a given timeframe. Additionally, handling API key management securely while optimizing usage across multiple agents added complexity to the system. Balancing computational overhead with these limitations required innovative solutions, such as implementing request batching and prioritization strategies. The pathway framework itself proved intricate, as designing a system capable of handling diverse datasets while maintaining consistency and quality demanded meticulous optimization and troubleshooting.

Key Learnings

The project highlighted the importance of agent-based modularity for flexibility and adaptability in research workflows. The integration of quality evaluation and aggregation steps proved critical for ensuring the reliability of the outcomes. Moreover, the dynamic nature of multi-agent systems showcased their potential as scalable solutions for complex decision-making pipelines, offering valuable insights into optimizing automated processes.

Future Scope

The pipeline has significant potential for enhancement, particularly in the integration of advanced models like SciBERT and other transformer-based architectures to compare the **LaTeX-formatted report structures** of conference papers with research papers. By leveraging these models, the system could quantitatively analyze the textual content, structure, and thematic alignment between conference submissions and previously published work. This approach would provide a deeper understanding of trends, gaps, and overlaps in research domains, while also addressing format-specific nuances inherent to LaTeX documents.

However, integrating such models into the pipeline presents challenges. **SciBERT** and similar models require extensive computational resources, which can strain the system, especially when processing large datasets of research and conference papers. Additionally, aligning their outputs to fit within the existing decision-making framework demands careful design of data preprocessing and model adaptation steps.

Quantitative metrics like structural coherence scores, Jaccard similarity, or cross-attention weight analysis can be introduced to evaluate how closely a paper matches the format and expectations of high-profile conferences. Although integrating these models presents computational challenges, especially when handling large datasets, solutions like model pruning, distributed inference, or optimized tokenization strategies could be adopted. Incorporating such approaches will significantly improve the system's ability to provide accurate, data-driven recommendations while aligning research outputs with established academic standards.