Degree Project in Computer Science and Engineering

Second cycle, 30 credits

AcademicRAG: Knowledge Graph Enhanced Retrieval-Augmented Generation for Academic Resource Discovery

Enhancing Educational Resource Discovery through Knowledge Graph-Based RAG Framework

SHUHUA CHEN

AcademicRAG: Knowledge Graph Enhanced Retrieval-Augmented Generation for Academic Resource Discovery

Enhancing Educational Resource Discovery through Knowledge Graph-Based RAG Framework

SHUHUA CHEN

Master’s Programme, Machine Learning, 120 credits
Date: June 3, 2025

Supervisors: Elias Zea, Shiva Sander Tavallaey, Fredrik Heintz
Examiner: Olov Engwall
School of Electrical Engineering and Computer Science
Swedish title: AcademicRAG: Knowledge Graph Enhanced Retrieval-Augmented Generation för upptäckt av akademiska resurser
Swedish subtitle: Förbättrad upptäckt av utbildningsresurser genom kunskapsgrafsbaserat RAG-ramverk

Abstract

Academic information retrieval faces significant challenges due to the complex semantic relationships inherent in scholarly knowledge. While Large Language Models have shown promise in processing academic content, they suffer from hallucinations. Retrieval-Augmented Generation (RAG) approaches mitigate this issue by grounding responses in external knowledge, but traditional RAG systems still lack the structural representation needed to capture intricate relationships between academic concepts.

This thesis presents AcademicRAG, a novel GraphRAG framework specifically designed for academic contexts. The framework integrates knowledge graph structures with language models to enhance semantic search capabilities through two key innovations: a clue-guided keyword generation method that reduces hallucinations and improves retrieval precision, and a subgraph-based local information extraction technique that captures deeper contextual relationships. Unlike popular GraphRAG approaches that rely on computationally expensive community structures, our framework uses hierarchical keywords for information retrieval, significantly reducing resource consumption while maintaining accuracy.

We developed a three-tiered data pipeline architecture that efficiently processes both unstructured and semi-structured academic texts, preserving semantic relationships while enabling incremental knowledge integration without requiring complete database reconstruction. The pipeline combines graph databases for relationship storage, vector databases for semantic embeddings, and key-value databases for document management, creating a robust foundation for academic knowledge discovery.

Evaluation on the UltraDomain dataset demonstrates that AcademicRAG consistently outperforms state-of-the-art frameworks across multiple dimensions, including comprehensiveness, diversity, and user empowerment. To validate practical utility, we implemented a Research Literature Assistant with specialized prompt and chunking function for academic papers. The assistant efficiently processes scholarly literature through a two-phase approach: first extracting abstracts and introductions to identify relevant papers and research trends, then enabling in-depth analysis of selected papers. This application performed exceptionally well in both simulated testing and user evaluations, providing contextually relevant insights and meaningful connections between academic resources.

AcademicRAG advances the field of academic information retrieval by enabling more intuitive, accurate, and efficient knowledge discovery, supporting researchers, educators, and students in navigating complex scholarly relationships.

Keywords

Information Retrieval, Large Language Models, Retrieval Augmented Generation, Knowledge Graph

Sammanfattning

Akademisk informationsinhämtning står inför stora utmaningar på grund av de komplexa semantiska relationer som finns i akademisk kunskap. Även om stora språkmodeller har visat sig vara lovande när det gäller att bearbeta akademiskt innehåll, lider de av hallucinationer. Retrieval-Augmented Generation (RAG)-metoder mildrar detta problem genom att grunda svar i extern kunskap, men traditionella RAG-system saknar fortfarande den strukturella representation som krävs för att fånga invecklade relationer mellan akademiska begrepp.

I den här avhandlingen presenteras AcademicRAG, ett nytt GraphRAG-ramverk som är särskilt utformat för akademiska sammanhang. Ramverket integrerar kunskapsgrafstrukturer med språkmodeller för att förbättra semantiska sökfunktioner genom två viktiga innovationer: en ledtrådsstyrd nyckelordsgenereringsmetod som minskar hallucinationer och förbättrar hämtningsprecisionen, och en subgrafbaserad lokal informationsextraktionsteknik som fångar djupare kontextuella relationer. Till skillnad från populära GraphRAG-metoder som förlitar sig på beräkningsmässigt dyra samhällsstrukturer, använder vårt ramverk hierarkiska nyckelord för informationshämtning, vilket avsevärt minskar resursförbrukningen samtidigt som noggrannheten bibehålls.

Vi utvecklade en tredelad datapipelinearkitektur som effektivt bearbetar både ostrukturerade och halvstrukturerade akademiska texter, bevarar semantiska relationer och möjliggör inkrementell kunskapsintegration utan att kräva fullständig databasrekonstruktion. Pipelinen kombinerar grafdatabaser för relationslagring, vektordatabaser för semantiska inbäddningar och nyckelvärdesdatabaser för dokumenthantering, vilket skapar en robust grund för akademisk kunskapsupptäckt.

Utvärdering på UltraDomain-datasetet visar att AcademicRAG konsekvent överträffar de senaste ramverken i flera dimensioner, inklusive fullständighet, mångfald och användarinflytande. För att validera den praktiska nyttan implementerade vi en forskningslitteraturassistent med specialiserad prompt- och chunkingfunktion för akademiska artiklar. Assistenten bearbetar effektivt vetenskaplig litteratur genom ett tvåfasigt tillvägagångssätt: först extraheras sammanfattningar och inledningar för att identifiera relevanta artiklar och forskningstrender, och sedan möjliggörs en djupgående analys av utvalda artiklar. Denna applikation presterade exceptionellt bra i både simulerade tester och användarutvärderingar, vilket gav kontextuellt relevanta insikter och meningsfulla kopplingar mellan akademiska resurser.

AcademicRAG utvecklar området akademisk informationssökning genom att möjliggöra mer intuitiv, korrekt och effektiv kunskapsinhämtning, vilket hjälper forskare, lärare och studenter att navigera i komplexa vetenskapliga relationer.

Nyckelord

Informationshämtning, Stora språkmodeller, Retrieval Augmented Generation, Kunskapsgraf

Acknowledgments

I wish to express my gratitude to my parents and family for their support throughout my Master's studies.

I sincerely thank my supervisors, Shiva Sander Tavallaey, Elias Zea, and Fredrik Heintz, for their valuable guidance and feedback on this thesis.

Special thanks to Zhuchenyang Liu for the productive collaboration that made this project possible.

I would like to extend my appreciation to SUPR (Swedish National Infrastructure for Computing) for providing the computational resources that were essential for this research.

I am also grateful to Dr. Yunfan Gao for permitting the use of the figure from his paper in this thesis.

Stockholm, June 2025
Shuhua Chen

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

Graph-based RAG	A technique that combines Retrieval-Augmented Generation with knowledge graphs to enhance information retrieval and generation capabilities.
KG	Knowledge Graph
LLM	Large Language Model
RAG	Retrieval-Augmented Generation

Chapter 1

Introduction

In this thesis project, we will develop a versatile Graph-based Retrieval-Augmented Generation (Graph-based RAG) framework for semantic search and discovery in academic contexts. Then, the developed framework will be applied to some specific academic scenes, such as the course syllabus exploration and literature discovery assistant.

1.1 Background

The rapid expansion of academic information in digital repositories has created both opportunities and challenges for researchers, educators, and students. While vast amounts of knowledge are now accessible, effectively navigating these resources remains problematic. This research addresses this challenge through the development of a Graph-based RAG framework specifically designed for semantic search and discovery in academic contexts.

Traditional information retrieval systems rely heavily on keyword matching, which often fails to capture the complex semantic relationships between academic concepts [1]. This limitation leads to incomplete search results and missed connections across disciplines. These systems particularly struggle with contextual understanding, synonym recognition [1], and identifying conceptual relationships that extend beyond simple word matching [2]. As a result, valuable academic resources often remain undiscovered, hindering knowledge transfer and interdisciplinary collaboration.

Recent advances in Large Language Models (LLMs) have significantly improved text processing capabilities. However, standard Retrieval-Augmented Generation (RAG) approaches still face limitations when applied to academic contexts. Conventional RAG methods enhance LLMs with external knowledge retrieval but typically use flat document structures that cannot adequately represent the complex relationships inherent in academic knowledge [3]. Academic information is characterized by intricate connections—such as prerequisite relationships between courses, citation networks across research papers, and conceptual hierarchies within disciplines—that are essential for comprehensive resource discovery.

Current research in information retrieval has begun exploring graph-based approaches for representing knowledge [4, 5, 6, 7, 8], but few studies have specifically addressed the unique challenges of academic contexts. Existing graph-based systems often focus on general knowledge domains or commercial applications [6, 4], lacking the customized designs needed for academic resource discovery. Additionally, studies examining the integration of knowledge graphs with language models have primarily concentrated on improving factual accuracy rather than enhancing contextual understanding and relationship navigation.

The integration of Knowledge Graph (KG) structures with RAG represents a promising yet underexplored approach to addressing these limitations. By explicitly modeling the relationships between academic entities and concepts, graph-based retrieval systems can potentially capture the multidimensional nature of academic knowledge more effectively than traditional vector-based approaches. This research direction aligns with recent trends toward more structured knowledge representation in artificial intelligence systems [4, 5, 7, 9, 10, 11, 8], while addressing the specific needs of academic information discovery.

1.2 Problem

1.2.1 Original problem and definition

This project explores the intersection of KGs, retrieval systems, and language models through three principal research questions:

1. RQ1: How should graph databases and language models be optimally integrated in a Graph-based RAG architecture to enable efficient academic resource discovery?

2. RQ2: What architectural patterns enable a flexible data pipeline capable of processing both unstructured and semi-structured academic content while preserving semantic relationships?

3. RQ3: How can the Graph-Based framework be effectively applied to downstream academic tasks, and what is its performance in real-world applications?

1.2.2 Scientific and engineering issues

Developing an effective Graph-based RAG framework tailored for academic text presents several challenges. These challenges stem from the absence of standardized evaluation datasets, the need for domain-specific adaptation, and the trade-off between computational efficiency and model performance. Addressing these issues is essential to ensuring the reliability, applicability, and scalability of the proposed framework.

- One of the primary challenges is assessing the Graph-based RAG framework. There is not a well-established dataset specifically designed for evaluating RAG systems with graph knowledge integration. Current benchmarks for RAG systems primarily focus on generic or domain-specific question-answering tasks but do not explicitly evaluate the benefits of incorporating structured KGs. Therefore, a critical aspect of this research involves designing a robust evaluation protocol that effectively measures the framework's performance.

- Another major challenge is ensuring that the Graph-based RAG framework is well-suited for processing and retrieving information from academic texts. Unlike general-domain text, academic literature is often dense, technical, and rich in domain-specific terminology. Standard language models may struggle with disambiguating complex concepts, and naive retrieval methods might fail to capture implicit relationships between entities.

- Integrating LLMs within the Graph-based RAG framework introduces a trade-off between computational resource consumption and overall system performance. On one hand, larger models with higher computational costs tend to achieve better generation quality and reasoning capabilities. On the other hand, excessive computational demands can hinder the framework's scalability, especially when dealing with large-scale academic corpora.

1.3 Purpose

The purpose of this project is the development of a versatile Graph-based RAG framework for semantic search and discovery in academic contexts. The framework should address the limitations of both traditional search systems and standard RAG implementations by explicitly modeling the interconnected nature of academic knowledge, enabling more intuitive navigation and discovery of resources. This framework can be applied to specific areas within academia, such as literature research assistants and course discovery systems, aiming to enhance the efficiency and accuracy of academic resource exploration.

The framework is designed for seamless adaptation to various downstream tasks. For the literature research assistant application, it offers a sophisticated stepwise literature search capability. This feature allows researchers to selectively analyze specific sections of papers rather than processing entire documents at once. Through this progressive methodology, users can initially examine introductions or abstracts from a broader collection of publications, select the most relevant sources, and subsequently perform detailed analysis on this carefully curated subset. This approach substantially decreases both cognitive demands and computational requirements when navigating extensive academic literature collections.

If the proposed Graph-based RAG framework successfully achieves its goals, several groups within academia will benefit significantly:

- Researchers and Academics: Traditional search engines often return results based on keyword matching, which can lead to an overwhelming number of irrelevant papers. By explicitly modeling relationships between concepts, the Graph-based RAG framework enables more semantically meaningful search results.

- Students and Educators: The framework can support course recommendation systems that go beyond simple keyword-based matching so students can find courses aligned with their knowledge level and learning goals more effectively. Educators can leverage the system to identify gaps or redundancies in academic programs by analyzing how topics and courses are interrelated within a KG.

- Academic Institutions and Digital Libraries: University libraries and online academic repositories can integrate the framework to provide context-aware document retrieval, improving access to research materials for students and faculty.

- Developers and AI Researchers: The Graph-based RAG framework provides a foundation that can be customized for various academic applications, from scientific paper recommendations to domain-specific question-answering systems. This project can also serve as a benchmark for further advancements in graph-based retrieval-augmented generation.

By enabling more intuitive, accurate, and efficient knowledge discovery in academic contexts, this framework has the potential to transform the way information is accessed, linked, and utilized in research and education.

The project may raise potential sustainability questions. One thing is that using LLMs may consume lots of energy. We will try to mitigate this problem by using models with fewer parameters and improving our framework to reduce model usage. Besides, perhaps LLMs will output some biased answers while we can tackle this ethical issue by modifying the prompt or filtering answers based on some rules.

1.4 Goals

The goal of this project is to develop a Graph-based RAG framework that is suitable for academic contexts. This has been divided into the following three sub-goals:

1. Subgoal 1: Design and implement a Graph-based RAG framework that effectively integrates graph databases and language models for academic resource discovery.
2. Subgoal 2: Develop a flexible data pipeline capable of processing unstructured and semi-structured academic content while preserving semantic relationships.
3. Subgoal 3: Apply the framework and pipeline to specific downstream tasks. Evaluate the performance of the Graph-based RAG system in practical applications and analyze its adaptability to domain-specific requirements with minimal modifications.

The deliverables and results of the project should include a versatile Graph-based RAG framework and its adaptation for real-world applications.

1.5 Division of the work

Given that a master's thesis is attributed to a single author, the jointly completed sections naturally resulted in similar content in the initial parts of our respective papers. To uphold academic rigor and meet the requirements of thesis writing, we have clarified the division of writing responsibilities for these shared sections. The details of this division are presented in Table 1.1.

Table 1.1: Division of the Labour for the Report Writing

Chapter	Section	Contributor(s)
Introduction	All Sections	S.C.
Background	2.1 Large Language Models	S.C., Z.L.
	2.2 Knowledge Graph	S.C., Z.L.
	2.3 Retrieval Augmented Generation	S.C., Z.L.
	2.4 Graph-Based Retrieval Augmented Generation	S.C., Z.L.
	2.5 Related works	S.C., Z.L.
	2.6 Summary	S.C.
AcademicRAG Framework	3.1 Overview of the framework	Z.L., S.C.
	3.2 Graph-based text indexing	Z.L., S.C.
	3.3 Graph-Guided query retrieval	S.C., Z.L.
	3.4 Graph-Enhanced Generation	Z.L., S.C.
	3.5 AcademicRAG Framework Evaluation	S.C., Z.L.
	3.6 Summary	Z.L., S.C.
Data Pipeline for AcademicRAG	4.1 Document Process Flow	S.C., Z.L.
	4.2 Document Index Flow	S.C., Z.L.
	4.3 Element Extraction Flow	S.C., Z.L.
	4.4 Data Pipeline Evaluation	Z.L., S.C.
	4.5 Summary	Z.L., S.C.
Application	All Sections	S.C.
Conclusions and Future Work	All Sections	S.C.

Note: Z.L. indicates the cooperator Zhuchenyang Liu (zhuchenyang.liu@aalto.fi) [12]. Names in front indicate the main author; names in back indicate the person responsible for refinement.

1.6 Research Methodology

1.6.1 Hypothesis

In this project, the following hypothesis will be examined:

1. The integration of graph database capabilities with language models will provide more accurate and contextually relevant academic resource discovery compared to traditional retrieval augmented methods, by leveraging both structural relationships and semantic understanding.

2. A modular pipeline architecture with specialized processors for different content types will effectively handle diverse academic materials while maintaining relationship integrity and semantic meaning across the system.

3. The applications of research literature assistant and course discovery systems will demonstrate different optimal configurations of the Graph-based RAG system, with research literature assistants benefiting more from citation network analysis and semantic similarity, while course discovery systems will rely more heavily on prerequisite relationships and topic clustering.

1.6.2 Evaluation of the Graph-based RAG Framework

The primary objective of the evaluation is to precisely measure the performance of the developed Graph-based RAG framework. Given the absence of a dedicated dataset for Graph-based RAG evaluation, we will adopt an approach similar to that of [4]. Specifically, we will generate a set of questions based on the text data, use the proposed framework to answer these questions, and then employ LLMs to compare the responses generated by different frameworks based on predefined evaluation criteria.

The baseline frameworks for comparison will include Graph-based RAG from [4], LightRAG [5] and RAG from [13], as they represent typical and well-established approaches in the relevant research domain. To ensure a robust and domain-relevant evaluation, we will utilize a subset of the UltraDomain dataset [7], which consists of textbooks from various academic fields. This dataset is particularly well-suited for assessing our framework's performance in academic context, as it covers a broad range of specialized knowledge. Additionally, the inherent length of textbooks can help evaluating the framework's ability to capture and leverage long-range relationships within the text, a key challenge in knowledge retrieval and reasoning.

1.6.3 Evaluation of the Data Pipeline

The data processing pipeline is designed to incrementally refine diverse academic content, ensuring structured and meaningful extraction of information. To assess our data pipeline's effectiveness, we will create a test dataset of interconnected academic documents with established relationships, such as citation networks between research papers and prerequisite chains among courses. After processing these documents and constructing the knowledge graph, we will use visualization techniques to analyze the extracted entities and relationships. We will assess several key aspects of the knowledge graph: (1) entity extraction completeness, (2) relationship identification accuracy, (3) preservation of document interconnections, and (4) semantic relevance of the extracted knowledge. This approach allowed us to evaluate how well the pipeline preserves semantic connections, identify potential extraction errors, and assess the overall structural integrity of the resulting knowledge graph. By focusing on document correlation rather than isolated extraction metrics, we can better understand the pipeline's capability to support the academic knowledge discovery goals of our framework.

1.6.4 Application in Real-World Scenarios

To validate the practical utility of our framework, we will apply the AcademicRAG system to two distinct real-world academic contexts: a research literature assistant and a course discovery system. This deployment allows us to assess the framework's adaptability and performance in addressing specific domain challenges.

The implementation process will verify whether our framework requires only minimal modifications when applied to different downstream scenarios. By maintaining the core architecture while adjusting only domain-specific components, we can evaluate the framework's versatility and generalizability across academic applications.

Our evaluation methodology combines simulated scenario testing and real user interactions. The simulated scenarios provide controlled conditions to measure system performance across predefined features. Real user testing complements this approach by capturing qualitative aspects including user satisfaction and perceived usefulness.

1.7 Delimitations

1.7.1 Development of the Framework

The core objective of this project is to adapt Graph-based RAG for academic applications. Rather than building the framework entirely from scratch, which would be both inefficient and unnecessary, we will leverage existing open-source Graph-based RAG framework as a foundation. Specifically, we will use LightRAG [5] as the base framework. The framework will be modified and optimized to better suit academic contexts, ensuring effective integration of structured knowledge and enhanced retrieval capabilities for scholarly content.

1.7.2 Scope of the Project

The scope of this research encompasses the design, implementation, and evaluation of the proposed framework, with particular focus on two practical applications: academic literature navigation and university course discovery. While the framework aims to be adaptable across various academic contexts, this thesis limits its experimental validation to these two use cases. Besides, the project will only focus on academic texts, all other academic resources like pictures, videos, slides will not be considered. The research addresses both technical aspects and practical considerations. The evaluation will measure both system performance and user experience to provide a comprehensive assessment of the framework in real-world academic scenarios.

1.7.3 Large Language Models

In this project, we employ pretrained LLMs such as Llama, Qwen and GPT. All models maintain a parameter count under 100 billion, striking an optimal balance between performance and computational demands. Smaller parameter models offer significant cost advantages when using API services, while open-source alternatives like Llama and Qwen enable flexible local deployment without heavy GPU needs. The reduced parameter count also ensures faster inference speeds—a critical advantage in our framework that requires extensive entity and relationship extraction from academic texts. This alternative represents a deliberate optimization of computational resources against response quality, particularly important for academic information retrieval where both accuracy and efficiency matter. By constraining model size, we further align with sustainability considerations, minimizing energy consumption during both development and deployment phases while maintaining the framework's effectiveness for academic knowledge discovery tasks.

1.8 Structure of the thesis

This thesis is organized into six chapters that systematically present our research on Graph-based RAG for academic contexts.

Chapter 2 provides the necessary background knowledge on LLMs, RAG, KGs, and Graph-based RAG approaches. This chapter establishes the theoretical foundation and contextualizes our work within existing research.

Chapter 3 presents our proposed AcademicRAG framework. It begins with an overview of the framework architecture, followed by detailed explanations of each component: graph-based text indexing, graph-guided query retrieval (including subgraph retrieval and clue-guided keyword generation methods), and graph-enhanced generation. The chapter concludes with a comprehensive evaluation of the framework's performance compared to existing approaches.

Chapter 4 describes the data pipeline developed for the AcademicRAG framework. It explains the three main workflows: document process flow, document index flow, and element extraction flow. The chapter also presents evaluation results that demonstrate the pipeline's effectiveness in processing academic content while preserving semantic relationships.

Chapter 5 demonstrates the practical application of our framework through a research literature assistant. This chapter discusses the specific adaptations made for this application and presents findings from both simulated literature review scenarios and real-world user testing.

Chapter 6 concludes the thesis by summarizing our contributions, analyzing the limitations of the current approach, and suggesting directions for future research. This final chapter provides a reflective assessment of the project and its implications for academic knowledge discovery.

Chapter 2

Background

2.1 LLMs

LLMs are a class of deep learning models trained on vast amounts of textual data to understand and generate human-like text. Built primarily on the Transformer architecture [14], modern LLMs leverage self-attention mechanisms to process sequential data while capturing long-range dependencies, enabling unprecedented performance in natural language processing (NLP) tasks.

The evolution of LLMs began with bidirectional models like BERT [15], which introduced masked language modeling for contextual representation learning. Subsequent auto-regressive models such as GPT [16] demonstrated the effectiveness of unidirectional transformers for text generation. The scaling of model parameters (e.g., GPT-3 with 175 billion parameters [17]) and dataset sizes revealed emergent capabilities, including few-shot learning and chain-of-thought reasoning [18].

Most advanced LLMs like GPT-4 [19] and Deepseek-R1 [20] are built on subsequent auto-regressive models and achieve state-of-the-art performance across diverse benchmarks, enabling applications ranging from conversational AI to code generation. The generation process can be formulated as the following optimization problem:

$$y_t = \arg\max_{y} \ \mathcal{P}_\theta(y_{t-1}, \cdots, y_1|x), \tag{2.1}$$

where $\mathcal{P}_\theta$ denotes the probability, x is the input and $y_1, \cdots, y_t$ are the output sequence of LLMs. LLMs have shown excellent performance in various fields such as writing, math and so on, even outperforming humans. However, LLMs face limitations including hallucination of factual inaccuracies [21], temporal knowledge cutoffs, and insufficient reasoning robustness [22].

Recent research focuses on alignment techniques like Reinforcement Learning from Human Feedback (RLHF) [23] and Group Relative Policy Optimization (GRPO) [24]. The development of open-source alternatives (e.g., LLaMA [25]) has further democratized LLMs research while raising discussions about ethical deployment.

2.2 KG

A KG is a structured representation of real-world entities and their relationships, organized as a directed graph $G = (E, R, T)$, where E denotes entities (e.g., people, concepts, or events), R represents relation types (e.g., "is-a," "part-of," or "located-in"), and $T \subseteq E \times R \times E$ is a set of triples [26]. Each triple $(h, r, t) \in T$ encodes the fact that a head entity h is connected to a tail entity t via relation r. For example, (Paris, capitalOf, France) captures geopolitical knowledge. This graph-based structure allows for efficient reasoning, querying, and integration of heterogeneous data sources, making KGs indispensable for semantic understanding and decision-making systems.

Building a KG involves extracting knowledge from unstructured text (via Named Entity Recognition and Relation Extraction) and structured data integration (e.g., Wikidata [27]), and completing existing KGs through link prediction or rule-based reasoning. For example, KnowEdu, an education-focused KG system, leverages neural sequence labeling and probabilistic association rules to derive pedagogical concepts and their prerequisite relationships from curriculum data [28]. Specially, because LLMs show excellent performance on semantic understanding and entities recognition, [4] builds the KG by utilizing LLMs to extract entities and relationships.

KGs have found extensive applications across various domains, leveraging their ability to model complex relationships and integrate heterogeneous data sources:

- Search Engines and Information Retrieval: Commercial systems like Google's KG leverage entity relationships to disambiguate queries and surface contextual information, improving search relevance a lot in benchmark evaluations.

- Recommendation Systems: KGs power recommendation engines by identifying connections between user preferences and available content, leading to more personalized suggestions.

- Healthcare and Life Sciences: KGs integrate vast amounts of biomedical data, aiding in drug discovery, disease research, and personalized medicine by uncovering hidden relationships among biological entities.

Emerging applications combine KGs with LLMs for enhanced reasoning and generation. DRAGON [29] achieved state-of-the-art performance on complex QA benchmarks including CSQA, OBQA [30, 31] and so on by training LLMs on KG data. Future directions include dynamic KG construction from multi-modal inputs and quantum-enhanced graph embeddings for real-time decision systems.

2.3 RAG

RAG is a hybrid approach that combines the power of pre-trained LLMs with external knowledge retrieval mechanisms to enhance performance on knowledge-intensive tasks [13, 3]. Unlike traditional language models that rely solely on parametric memory—where knowledge is stored within model weights—RAG incorporates non-parametric memory by retrieving relevant documents from an external database before generating responses. This integration mitigates the common limitations of LLMs, such as hallucinations, outdated knowledge, and limited interpretability, making RAG a critical advancement in natural language processing (NLP).

The traditional RAG framework consists of three primary steps:

- Step 1. Indexing: The first step involves preparing and structuring raw data from diverse formats, such as PDF, HTML, Word, and Markdown. These documents are processed through a cleaning and extraction pipeline to ensure uniform plain-text representation. Given the context length limitations of LLMs, the text is segmented into smaller, manageable chunks. Each chunk is then encoded into a vector representation using an embedding model and stored in a vector database, enabling efficient similarity searches in the retrieval phase.

- Step 2. Retrieval: When a user submits a query, the system first encodes it into a vector representation using the same embedding model used during indexing. The RAG framework then computes similarity scores between the query vector and the stored text chunks in the database. The top K most relevant chunks are retrieved based on their similarity to the query, ensuring that the response is grounded in relevant external

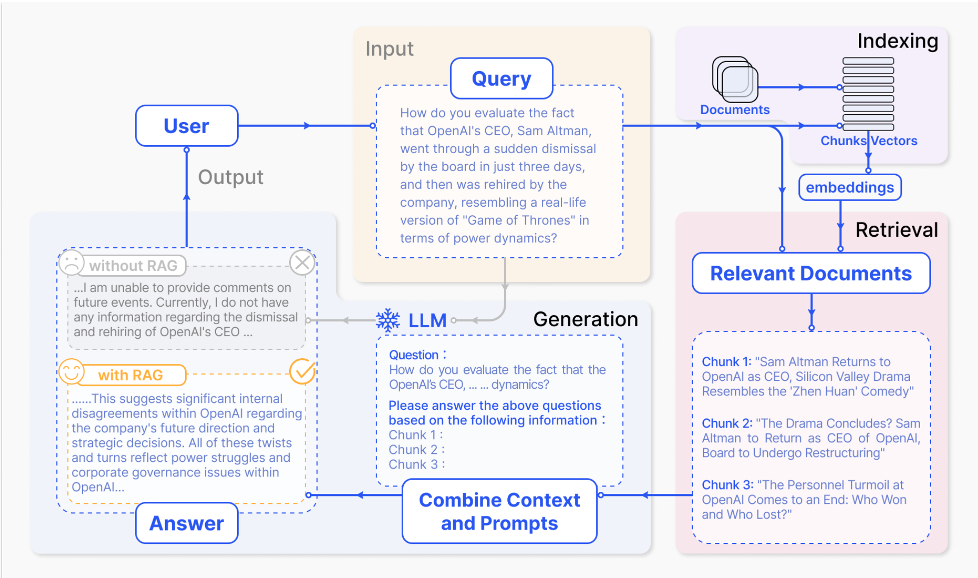

Figure 2.1: RAG process applied to Question Answering [3]
Note: This image is used with the permission of the author.

knowledge. These retrieved chunks are incorporated into the expanded context of the prompt.

- Step 3. Generation: In the final stage, the retrieved documents and the original query are combined into a structured prompt, which is then processed by an LLM, such as GPT [19]. The model generates a response based on the retrieved information while also leveraging its internal parametric knowledge when necessary. Depending on the application, the LLMs can be restricted to only use the retrieved documents to enhance factual accuracy. Additionally, in cases involving multi-turn dialogue, the conversation history can be incorporated into the prompt, allowing the model to maintain coherence across interactions.

The mechanisms of RAG enable LLMs to leverage external knowledge dynamically, improving their ability to generate accurate, informative, and up-to-date responses. Unlike standard LLMs that rely solely on memorized knowledge from pretraining, RAG retrieves relevant information from external sources, making it highly adaptable to rapidly evolving domains such as news and scientific literature. One of RAG's most significant advantages is its ability to mitigate hallucinations, a common issue where LLMs generate seemingly plausible but factually incorrect information. By grounding responses in retrieved, verifiable documents, RAG improves factual consistency and reliability. Additionally, RAG enhances interpretability and transparency, as users can inspect the external sources that contribute to the generated responses. A typical RAG application is illustrated in Figure 2.1. Consider a scenario where a user asks ChatGPT about a recently trending news topic. Since ChatGPT's knowledge is limited to its pretraining data, it may lack updated information on recent developments. The RAG framework bridges this gap by retrieving relevant news articles from external databases. These retrieved documents, combined with the original query, form a comprehensive prompt that enables the LLMs to generate a factually grounded and well-informed response.

Nowadays, RAG has been used in various fields: **(1) Open-Domain Question Answering**: RAG significantly improves performance on QA tasks by fetching relevant passages from Wikipedia or specific datasets. It has outperformed traditional extractive QA models on datasets such as Natural Questions (NQ) and TriviaQA [32]. **(2) Legal and Medical NLP**: Due to the necessity of precise and up-to-date knowledge, RAG has been applied in legal and medical domains [33, 34], where it retrieves case law or medical literature to assist decision-making. **(3) Chatbots and Virtual Assistants**: By integrating RAG, conversational agents such as ChatGPT can provide more accurate responses by retrieving real-time information, such as financial data, technical documentation, or recent news.

2.4 Graph-based RAG

Graph-based RAG is an extension of RAG that enhances retrieval mechanisms by incorporating structured knowledge from graphs, such as KGs and other relational databases [8]. Unlike traditional RAG systems that primarily retrieve and use unstructured text, Graph-based RAG retrieves structured entities, relationships, and subgraphs to enrich language model responses with contextual and relational knowledge. This approach enables more precise reasoning, improved factual accuracy, and enhanced interpretability, making it particularly useful in domains requiring strong logical consistency and entity relationships.

The Graph-based RAG framework can be decomposed into three primary stages:

- Step 1. Graph-Based Indexing (G-Indexing): This step involves the construction or selection of a structured graph database that serves as the retrieval corpus. Graph-based indexing facilitates efficient querying by organizing nodes (entities) and edges (relationships) in a way that optimizes search operations. The graph data can come from open KGs such as Wikidata [27] or be built from private datasets. Indexing constitutes the initial and the most important phase of Graph-based RAG as it determines the granularity of the subsequent retrieval stage, playing a crucial role in enhancing query efficiency.

- Step 2. Graph-Guided Retrieval (G-Retrieval): Instead of retrieving isolated text snippets, Graph-based RAG aims to retrieve the most relevant graph elements such as nodes, triples (subject-predicate-object), paths, or subgraphs relevant to a given query. Retrieval methods may include traditional graph traversal algorithms, semantic similarity searches, or graph neural network (GNN)-based retrieval models.

- Step 3. Graph-Enhanced Generation (G-Generation): Retrieved graph data is integrated into the response generation process. This can be done by converting graph structures into textual formats or directly using structured data representations through graph-aware transformers or hybrid models that combine GNNs with LLMs. In this stage, the generator takes the query, retrieved graph elements, and an optional prompt as input to generate a response.

These three stages work together to improve the contextual depth of generated responses, ensuring that relational knowledge is incorporated effectively.

Graph-based RAG enhances traditional RAG by integrating structured knowledge, leading to improved reasoning, accuracy, and interpretability. Compared to text-based retrieval, the incorporation of structured relational data brings several notable advantages: **(1) Preservation of Relational Knowledge**: Unlike RAG, which primarily focuses on retrieving semantically similar text, Graph-based RAG explicitly models relationships between entities. This structured retrieval enables more precise reasoning, especially for multi-hop inference tasks where multiple entities and their relationships need to be considered together. **(2) Efficient Knowledge Integration**: Graph structures enable a more hierarchical and contextual organization of knowledge, making it easier to integrate diverse sources of information into LLMs. Particularly, Graph-based RAG allows selective retrieval of only relevant knowledge (e.g., specific relationships rather than full documents), reducing noise and irrelevant information. **(3) Reduced Information Redundancy**: By retrieving subgraphs instead of long textual passages, Graph-based RAG minimizes irrelevant content and improves response efficiency. Since graphs store knowledge in an explicit, compressed format, Graph-based RAG also mitigate the "Lost in the Middle" problem [35] where transformers fail to attend to relevant context due to input length constraints.

Despite its advantages, Graph-based RAG still faces some challenges. Building and maintaining high-quality KGs require significant effort and domain expertise, which introduces extra burden. Besides, there is not a universal retrieval paradigm that can efficiently retrieving relevant subgraphs from large-scale graph databases, which remains an open research problem. Converting structured graph data into formats suitable for LLMs without losing relational integrity is also an ongoing challenge.

2.5 Related Work

2.5.1 Graph-Based Retrieval-Augmented Generation

Graph-based RAG extends standard RAG by leveraging structured KGs to enhance retrieval and reasoning. One of the pioneering approaches GraphRAG in this domain was introduced by Edge et al. [4], which employs LLMs to extract entities and relationships from unstructured text and constructs a KG. To facilitate efficient retrieval, Edge et al. [4] use community detection algorithms to partition the KG into distinct communities and then apply LLMs to generate community reports summarizing each community's key information. During retrieval, the model fetches the most relevant community reports to guide response generation. However, this approach has several key limitations: **(1) High computational cost**: Generating and storing community reports for all communities requires extensive computational resources. **(2) Scalability issues**: When new knowledge is inserted, all community reports must be recomputed, making it inefficient for dynamic and evolving datasets. **(3) Noise from irrelevant nodes**: Since the community reports summarize large portions of the KG, they often include unrelated nodes, introducing unnecessary noise into the retrieval process.

2.5.2 Improvements in Graph-based RAG

To reduce computational costs and noise, LightRAG proposes a dual-level retrieval framework that replaces communities and reports with local and global keyword extraction [5]. Instead of generating a community report for each partition, LightRAG extracts two levels of keywords: Local keywords, which capture entity-specific information. Global keywords, which provide a broader conceptual context. By indexing keywords instead of full community summaries, LightRAG significantly reduces retrieval latency and storage requirements while minimizing retrieval noise. Similar to LightRAG, our method extract local and global keywords during indexing step, but we further extract content keywords from each document chunk to build an independent keyword database. Medical GraphRAG optimizes GraphRAG for medical applications by integrating external domain-specific knowledge sources such as medical textbooks and controlled medical vocabularies [6]. Instead of relying on community detection, Medical GraphRAG categorizes graph nodes based on predefined medical terminology. Relying on terminology is a powerful classification method but it does not generalize well beyond the medical field. Instead, content keywords in our framework can be used universally for every domain.

2.5.3 Rule-Based Knowledge Graph Approaches for Literature

Unlike LLM-generated KGs, some methods, such as CG-RAG [36] and the scholarly KG framework by Jia et al. [37], construct KGs using manual rule-based techniques. These methods rely on explicitly defined citation relationships between academic documents and allow for precise, curated KGs that maintain citation integrity. However, these approaches have several drawbacks. They are effective for scientific papers and literature-based queries but do not extend well to other academic texts like course syllabus or textbooks. Besides, since the KG relies on manually inserted relationships, many implicit entity connections may remain undiscovered.

2.5.4 Graph Retrieval Mechanism

Edge et al.'s GraphRAG retrieves community reports, but this introduces unnecessary noise and includes many unrelated nodes, reducing retrieval precision. LightRAG retrieves entities and relationships based on keywords, reducing retrieval complexity. However, these nodes may lack direct connections, potentially missing critical relationships between entities. Additionally, because keywords are generated based on LLMs internal knowledge, this can introduce hallucinations. GRAG introduces ego-graph retrieval, where retrieval focuses on a local subgraph centered on an entity of interest [10]. While this approach improves retrieval relevance, it lacks strong inter-node connections and requires additional graph embedding models, increasing implementation complexity. Instead, our method first generates keywords based on the content keywords extracted from document, then retrieve relevant nodes and edges as well as the subgraph by using graph traversal method, which mitigates the hallucinations and captures some potential elements in graph.

2.6 Summary

LLMs have demonstrated remarkable performance across various NLP tasks. However, they suffer from a critical limitation—hallucination, where models generate plausible yet factually incorrect information. RAG addresses this issue by integrating external knowledge retrieval with LLMs internal knowledge. Despite its advantages, traditional RAG faces challenges such as overlooking long-range relationships and retrieving excessive or irrelevant information.

A promising solution to these limitations is integrating RAG with KGs. By leveraging structured knowledge representations, Graph-based RAG enables models to retrieve not only semantically relevant texts but also explicit relationships between entities, offering richer and more precise contextual information. This structured approach enhances LLMs reasoning and factual consistency, leading to more accurate and reliable text generation. Conversely, LLMs also contribute to KG construction, providing a new paradigm for automatically discovering potential entities and relationships in unstructured data, thereby enhancing the completeness and usability of KGs.

However, Graph-based RAG is still an evolving field, and existing methods exhibit various drawbacks. Community-based approaches [4] require high computational resources and suffer from scalability issues when handling large and dynamic input. Rule-based methods [36, 37] lack generalizability, making them unsuitable for diverse domains. Meanwhile, keyword-based techniques [5, 6] help reducing computational overhead and ego-graph retrieval improves local information extraction, but they still struggle with retrieval quality.

Building on these insights, we propose AcademicRAG, a Graph-based RAG framework specifically designed for academic settings. AcademicRAG aims to deliver high computational efficiency while ensuring accurate and contextually relevant knowledge retrieval, effectively balancing retrieval quality with computational cost. By addressing known limitations of current Graph-based RAG approaches, AcademicRAG provides a scalable solution adapted for academic research and information retrieval tasks. The proposed enhancements align closely with the core architectural principles and original design goals of existing frameworks. Finally, comprehensive empirical evaluations across multiple domains remain necessary to clearly assess the improvements in retrieval accuracy, computational efficiency, and overall system performance.

Chapter 3

AcademicRAG Framework

3.1 Overview of the Framework

AcademicRAG is a novel framework collaboratively developed to address the limitations of traditional retrieval systems in academic contexts. By integrating KG databases with RAG techniques, this framework creates a more semantically aware system capable of understanding and leveraging complex relationships inside the academic context.

Similar to other Graph-based RAG architectures [4, 6, 36, 10, 11, 8, 38, 39], AcademicRAG operates through three primary components, which support automatic graph database generation, subgraph extraction based on query statements and final answer generation, respectively.

The AcademicRAG framework, as illustrated in Figure 3.1, consists of two processing flows that work in conjunction to deliver contextually rich and semantically aware responses to academic queries.

The Index Flow, shown on the right side, focuses on the automatic graph database generation. It begins with document ingestion, followed by chunking to create manageable text segments. These chunks undergo entity and relationship extraction using LLMs, which identify key concepts, academic terms, and the connections between them. After extraction refinement to ensure accuracy, this structured information is inserted into a dual-database architecture: a Graph Database that preserves relational information between academic entities, and a Vector Database that stores semantic embeddings of content.

The Query Flow, depicted on the left, handles user interactions through a multi-stage process. It starts with natural language query input, which undergoes clues retrieval to identify key academic concepts. These concepts

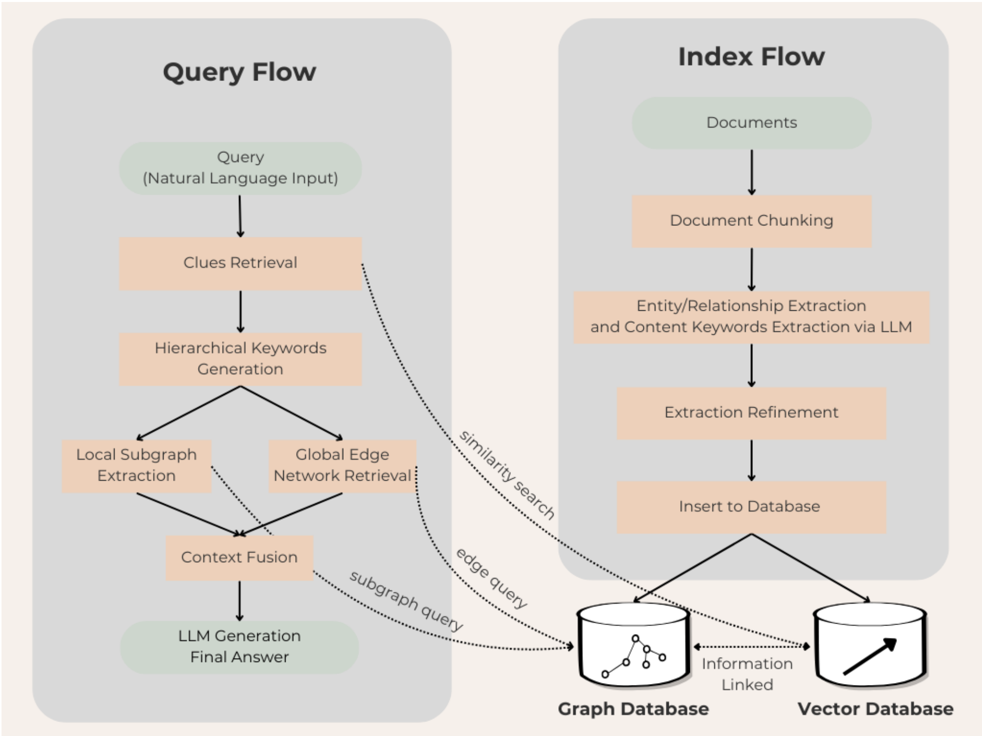

Figure 3.1: AcademicRAG Framework Flowchart

are then organized into hierarchical keywords, enabling two parallel retrieval paths: Local Subgraph Extraction, which retrieves relevant concept networks from the Graph Database, and Global Edge Network Retrieval, which identifies broader relationship patterns. As shown in Figure 3.2, the Context Fusion stage integrates these retrieval results, providing a comprehensive knowledge foundation for the final LLM-based answer generation.

In the following sections, we will examine each of these components in detail, exploring their design principles, implementation considerations, and specific adaptations for academic resource discovery. We will outline how these components work together to create a unified framework that enhances both the retrieval precision and contextual awareness of academic information systems.

3.2 Graph-based Text Indexing

Graph-based text indexing is the first component of the AcademicRAG framework, transforming unstructured academic documents into a structured,

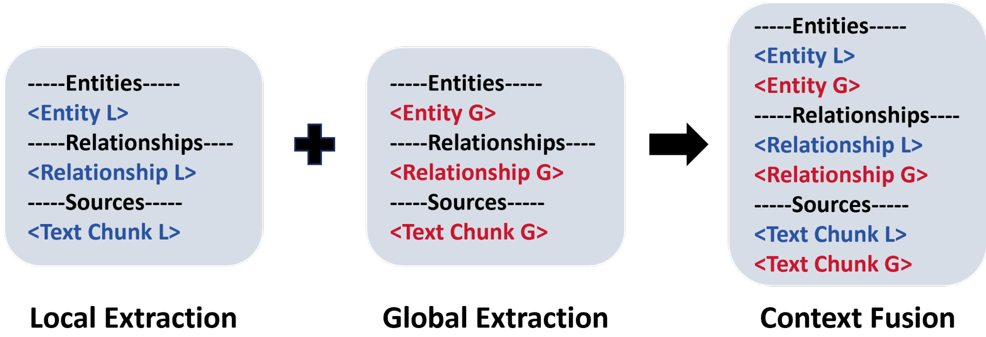

Figure 3.2: Illustration of Context Fusion

queryable knowledge representation. The process is meticulously designed to capture the semantic richness of academic content through three stages - document chunking, entity extraction and content keyword extraction.

3.2.1 Document Chunking

The initial stage of the indexing process involves segmenting academic documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ into text chunks $\mathcal{C} = \{c_i \mid c_i \subseteq d_j, \forall d_j \in \mathcal{D}\}$ for processing. In the entity extraction stage, these text chunks will be passed to LLMs with prompts designed to extract the various elements of a graph index. Our approach employs a fixed-size chunking strategy optimized for academic content, the chunk length $|c_i|$ is a hyperparameter that can be varied in different application scenes.

3.2.2 Entity Extraction

The second stage of the indexing framework involves the systematic extraction of entities and relationships from document chunks through the application of LLMs. This phase is structured to ensure precision and adaptability to diverse academic domains.

LLMs are deployed with domain-specific prompts engineered to optimize entity and relationship extraction from academic texts. The input prompts are meticulously designed to embed chunked content, thereby directing the model to generate contextually relevant entities and relationships. Entities are identified and classified according to a domain-adaptive schema, which can be tailored to accommodate discipline-specific terminologies and conceptual hierarchies. Relationships, represented as undirected edges between entities, are formulated as natural language statements that encapsulate semantic associations, such as causal links, functional dependencies, or categorical affiliations. Inspired by Wu et al. [6], who use external terminologies to work as global connections, we propose content keywords to capture the global themes of text units. During the extraction process, content keywords are extracted from each chunk. These content keywords refer to global level keywords that summarize the main concepts, themes, or topics of the entire text chunks. These should capture the overarching ideas present in the document.

To facilitate this extraction, we utilize a LLM specifically configured for entity-relationship extraction. This model operates based on a domain-specific prompt template, $P_{\text{domain}}(d)$, which simultaneously instructs both entity identification and relationship extraction. The prompt is tailored to a particular domain d, ensuring that the extraction process aligns with domain-specific requirements.

The entity extraction process can be formulated as follows:

$$(E_i, R_i, \mathcal{K}_i^{\text{content}}) = \text{LLM}_{\text{Extraction}}\left(P_{\text{domain}}(d), c_i\right), \ \forall i \tag{3.1}$$

where $\text{LLM}_{\text{Extraction}}$ is the LLM configured for extraction, c_i is an input document chunk, $E_i = \{e_1, e_2, \ldots, e_n\}$ is the set of all recognized entities within the chunk, $R_i = \{(e_j, r_{jk}, e_k) \mid e_j, e_k \in E_i\}$ represents relationships between entity pairs with r_{jk} connecting entities e_j and e_k, and $\mathcal{K}_i^{\text{content}}$ represents global-level terms that capture the core concepts and themes of the document chunk.

To enable efficient retrieval from the graph database, we utilize an LLM-powered function $F(\cdot)$ to generate key-value pairs (K, V) for each entity e_i and relationship r_i. Each index key is a word or short phrase that facilitates fast and relevant retrieval, while the corresponding value is a textual summary containing relevant snippets from external data to support knowledge-enhanced text generation. Entities use their names as unique index keys, whereas relationships may have multiple index keys, derived from LLM-enhanced representations that incorporate global themes from connected entities.

A multi-pass iterative refinement mechanism is employed to maximize entity and relationship extraction across data chunks. The text chunks $\{c_i\}_{i=1,2,\cdots}$ are repeatedly processed through multiple extraction rounds to capture as many entities and relations as possible. However, the multi-pass iterative extraction result in an exponential increase in the number of LLM calls. Therefore, this represents a tradeoff between performance and cost. Empirically, performing a single reextraction achieves the optimal balance between performance and cost.

During the extraction phase, duplicate entities are systematically consolidated to ensure the structural consistency of the graph. Concurrently, duplicate relationships are processed using LLM, which generate a synthesized and integrative description aimed at preserving both the integrity and completeness of relational information. This methodology minimizes redundancy while enhancing the overall coherence and semantic fidelity of the resulting KG.

The refined entities and relationships are persistently stored in a graph database, which preserves their structured interconnections and enables efficient traversal for query resolution. Concurrently, dense vector embeddings are generated for both entities and relationships using embedding models, and these embeddings, along with the extracted content keywords, are indexed in a vector database. This dual-storage architecture facilitates hybrid retrieval capabilities: the graph database supports structured queries based on explicit relationships, while the vector embeddings enable similarity-based searches, enhancing the system's ability to retrieve contextually relevant information. Another key advantage of our storage structure is its adaptability to incremental data updates. Unlike the community structure proposed in [4], which requires recomputing entire community structures and reports when the graph updates, our framework eliminates this overhead by leveraging key-value pairs and content keywords for information retrieval. When a new document is added, our indexing strategy ensures that only the newly extracted entities, relationships, and content keywords are incorporated into the database after deduplication, rather than re-generating all communities and reports. This design significantly enhances efficiency and scalability, making the framework more suitable for dynamic and continuously evolving knowledge bases.

By integrating LLM-driven extraction, iterative refinement, and dual-format storage, this approach establishes a robust foundation for advanced academic knowledge retrieval, ensuring high fidelity in entity-relationship mapping and scalability across heterogeneous scholarly corpora.

3.3 Graph-Guided Query Retrieval

Graph-guided query retrieval enables efficient and context-aware information retrieval by leveraging structured relationships and semantic embeddings. This process is divided into three key stages.

3.3.1 Keywords Retrieve Using Clues

Upon query initiation, the system employs a multi-stage retrieval strategy to optimize keyword relevance and contextual alignment. First, the content keywords vector database is queried to identify the content keywords that exhibit maximal semantic congruence with the user's query. These content keywords, synthesized from both the query and the indexed corpus, function as retrieval clues to guide subsequent processing. The clues are retrieved via the equation:

$$\mathcal{K}^{\text{clues}} = \{k \in \mathcal{K}^{\text{content}} \mid \text{sim}\left(f_{\text{embed}}(Q), f_{\text{embed}}(k)\right) \geq \tau\}, \tag{3.2}$$

where Q represents the user query in natural language form, while $\mathcal{K}^{\text{content}}$ denotes the content keyword vector database. The embedding model, $f_{\text{embed}}(\cdot)$, projects text into a vector space, enabling the computation of semantic similarity through a predefined metric, $\text{sim}(\cdot,\cdot)$, such as cosine similarity. A threshold parameter, τ, determines the minimum required semantic congruence for a keyword k to be included in the retrieved set $\mathcal{K}^{\text{clues}}$, which subsequently guides further processing.

An LLM is then invoked to generate a hierarchical keyword set, comprising high-level (conceptual/thematic) and low-level (specific/contextual) terms derived from the original query and the identified clues. Low-level keywords target localized information retrieval, prioritizing granular details within narrow contexts, while high-level keywords facilitate broad-scope exploration of conceptual relationships. The process can be written as:

$$(\mathcal{K}_h, \mathcal{K}_l) = \text{LLM}_{\text{keyword}}\left(Q, \mathcal{K}^{\text{clues}}\right). \tag{3.3}$$

$\mathcal{K}_h$ corresponds to high-level keywords that encapsulate conceptual or thematic aspects, whereas $\mathcal{K}_l$ comprises low-level keywords focusing on specific contextual details. The model $\text{LLM}_{\text{keyword}}$ is specifically designed to generate these dual-level keywords by leveraging both the original query Q and the retrieved content keywords $\mathcal{K}^{\text{clues}}$. Figure 3.3 illustrates the complete process for generating dual-level keywords. Content keywords are retrieved based on vector similarity with the user's query. These retrieved keywords serve as contextual clues and are input to LLMs along with the original query. The ?? then generate both high-level and low-level keywords. This dual-level keyword generation mechanism ensures adaptive alignment with user intent, enabling hybrid retrieval strategies that balance specificity and generality.

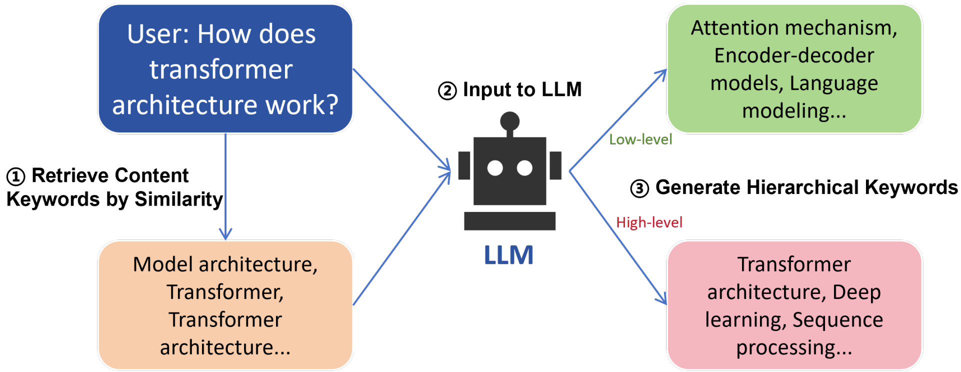

Figure 3.3: Example of Keywords Retrieve Using Clues

3.3.2 Local Information Extraction based on Subgraph

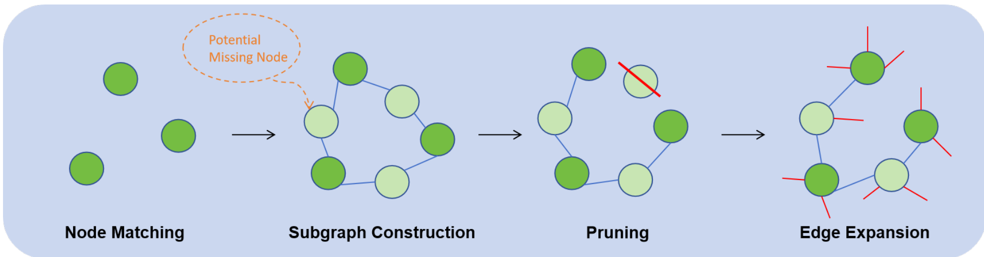

Figure 3.4: The Process of Subgraph Building

For queries requiring localized insights, our framework executes a structured subgraph extraction protocol driven by low-level keywords $\mathcal{K}_l$. The workflow is formalized as follows:

1. Node Matching: Graph nodes are matched against low-level keywords $\mathcal{K}_l$ using vector similarity metrics, prioritizing nodes with higher cosine similarity to the keyword embeddings.

2. Subgraph Construction: A subgraph is generated by identifying the shortest paths between matched nodes, ensuring topological coherence while preserving contextual proximity.

3. Pruning: Edges are filtered based on semantic relevance thresholds, retaining only those whose descriptive statements align with the query keywords. Some isolated nodes after filtering edges will also be removed.

4. Text Unit Extraction: Textual chunks that are associated with retained edges and appear in the nodes' one-hop ego-graph are ranked by frequency within the subgraph, with top-ranked segments prioritized for retrieval.

5. Edge Expansion: The subgraph is iteratively expanded by incorporating one-hop neighbors of matched nodes, introducing additional edges that satisfy relevance criteria.

6. Data Truncation: To adhere to computational constraints, all extracted data—including nodes, edges, and text units—are truncated iteratively until the total token count complies with predefined thresholds.

This pipeline outputs entities, relationships, and original text segments, ensuring that contextual fidelity is preserved while effectively capturing potential relationships. Figure 3.4 shows how the subgraph is retrieved. Unlike LightRAG [5], which restricts retrieval to one-hop neighbors, and GRAG [10], which relies on k-hop ego-graphs, our approach retrieves a complete subgraph for local information extraction, enabling the construction of richer connections between different nodes. This method enhances retrieval depth and knowledge integration, providing a more comprehensive representation. The whole local information extraction pipeline can be formulated as:

$$\mathcal{L}_{\text{local}} = (E_{\text{local}}, R_{\text{local}}, T_{\text{local}}) = \text{pipeline}_{\text{local}}(\mathcal{K}_l, \mathcal{G}), \tag{3.4}$$

where $\mathcal{L}_{\text{local}}$ represents the local information extracted, $\text{pipeline}_{\text{local}}$ denotes the local extraction function that processes low-level keywords $\mathcal{K}_l$ against the KG $\mathcal{G}$, producing a set containing local entities E_{local}, relationships R_{local}, and associated text units T_{local}.

3.3.3 Global Information Extraction

For broad-scope queries, the framework leverages high-level keywords $\mathcal{K}_h$ to retrieve global knowledge structures through the following steps:

1. Edge Retrieval: The vector database is queried to identify edges whose embeddings correlate with high-level keywords, prioritizing edges that encapsulate conceptual relationships (e.g., theoretical frameworks, domain-wide trends).

2. Nodes Retrieval: Nodes related to the retrieved edges are extracted and sorted by their degree.

3. Edge Data Processing: Retrieved edges are analyzed to extract metadata, including degree (sum of degree between source and target nodes) and weight (semantic connection strength), quantifying their structural and contextual significance.

4. Integration and Ranking: Edges are aggregated and ranked using a composite metric combining relevance scores (based on keyword alignment) and connection weights, ensuring that dominant conceptual linkages are prioritized.

5. Text Unit Extraction: Text chunks that are associated with edges are ranked by the similarity between edges and high-level keywords and retrieved.

6. Data Truncation: To adhere to computational constraints, all extracted data—including nodes, edges, and text units—are truncated iteratively until the total token count complies with predefined thresholds.

The final output includes interconnected entities, relationships, and supporting textual evidence, enabling holistic knowledge discovery across academic domains. By incorporating high-level keywords derived from content keywords, our framework effectively retrieves global information from the graph without relying on the community structure used in [4]. This approach optimizes retrieval efficiency while maintaining a balance between performance and computing resource consumption, providing a scalable and efficient solution. The whole global information extraction pipeline can be formulated as:

$$\mathcal{L}_{\text{global}} = (E_{\text{global}}, R_{\text{global}}, T_{\text{global}}) = \text{pipeline}_{\text{global}}(\mathcal{K}_h, \mathcal{G}) \tag{3.5}$$

Where $\mathcal{L}_{\text{global}}$ represents the global information extracted, $\text{pipeline}_{\text{global}}$ denotes the global extraction function that processes high-level keywords $\mathcal{K}_h$ against the KG $\mathcal{G}$, producing a set containing global entities E_{global}, relationships R_{global}, and associated text units T_{global}.

3.4 Graph-Enhanced Generation

The final stage of the AcademicRAG pipeline synthesizes localized and global information into a coherent, contextually grounded response. Local subgraphs, extracted via low-level keywords, and global relationship networks, retrieved via high-level keywords, are concatenated into a unified contextual framework. This integration combines dual-level entities, relationships, and source text anchors. Eventually, this integrated information is embedded in a prompt predefined for academic texts, generating the final answer through a large model. The graph-enhanced generation process is shown as Eq.3.6:

$$A_{\text{final}} = \text{LLM}_{\text{response}}\Big(P_{\text{final}},\ \text{Concat}\big(\mathcal{L}_{\text{local}}, \mathcal{L}_{\text{global}}\big)\Big), \tag{3.6}$$

where $\text{LLM}_{\text{response}}$ refers to the LLM configured to generate the final answer based on the given input. P_{final} represents the prompt used to format the input for the model. The information extracted by low-level keywords is denoted as $\mathcal{L}_{\text{local}}$, while information retrieved by high-level keywords is denoted as $\mathcal{L}_{\text{global}}$. The $\text{Concat}(\cdot, \cdot)$ operator performs the contextual fusion of these dual-level components, combining entities, relationships, and source text anchors into a cohesive structure. Finally, the final answer A_{final} is generated after this retrieval and synthesis process.

3.5 AcademicRAG Framework Evaluation

3.5.1 Experiment Settings

To assess its performance in an academic context, we utilize a subset of the UltraDomain benchmark [7] as the evaluation dataset. This dataset comprises 428 college textbooks spanning 18 academic domains, providing a diverse and comprehensive knowledge base. For our experiments, we focus on two specific domains—agriculture and computer science—which include a total of 22 college textbooks (10 from computer science and 12 from agriculture). The considerable amount of textbooks ensures that the dataset can cover enough domain knowledge to conduct the query test and ensure the validity of the evaluation.

To assess the effectiveness of RAG systems in high-level sensemaking tasks, we aggregate all textual content from each domain as context and adopt the generation methodology described in [4]. Specifically, we first instruct an LLM to generate five distinct RAG users, each representing a unique academic user, such as a ”PhD Student Specializing in NLP” or a ”Farmer Transitioning to Organic Practices.” This approach allowed us to capture diverse information needs across the academic spectrum. Each synthetic user was assigned five distinct tasks that reflected common academic activities, such as ”Identify prerequisite courses for ML” or ”Adopt soil-building techniques without synthetic inputs”. These tasks were deliberately designed to emphasize different cognitive intents, including exploratory, comparative, and problem-solving dimensions. For question synthesis, we generated five questions for each user-task pair, resulting in a total of 125 questions per domain (5 users × 5 tasks × 5 questions). These questions were crafted to require holistic understanding of the corpus rather than simple fact retrieval.

To establish the effectiveness of our approach, we compared AcademicRAG against the following state-of-the-arts models on our evaluation dataset:

- NaiveRAG [13]: This model serves as a standard baseline in existing RAG systems. It segments input contexts into smaller text chunks and stores them in a vector database using text embeddings. During retrieval, NaiveRAG directly embeds the user's query into a vector representation and retrieves the most relevant text chunks based on similarity scores, ensuring an efficient and straightforward retrieval process.

- GraphRAG [4]: This is a Graph-based RAG system that leverages LLMs to extract entities and relationships from input text, structuring them as nodes and edges within a graph database. The system further generates descriptions for these elements, clusters nodes into communities for retrieval, and produces community reports to encapsulate global contextual information. When handling queries, GraphRAG enhances retrieval quality by traversing communities, enabling access to more comprehensive and interconnected knowledge.

- LightRAG [5]: An adaptation of GraphRAG, LightRAG follows a similar approach by utilizing LLMs to construct graph-based representations. However, instead of clustering elements into communities, it employs keywords to classify them. When processing queries, LightRAG retrieves one-hop neighbors and their relationships by keywords, enabling faster response generation while maintaining relevant contextual knowledge.

All of the evaluated frameworks use the same configuration. For graph construction, we employed Qwen2.5-72B [40] to extract entities and relationships from the corpus, which were subsequently stored in the graph database. For the generation phase, we continue to employ Qwen2.5-72B to ensure consistent and relevant outputs. To maintain consistency across all datasets, the chunk size is uniformly set to 1200. Additionally, the gleaning parameter is fixed at 1 for GraphRAG, LightRAG, and our framework, ensuring a standardized retrieval process across different models.

Defining ground truth for those queries involving complex high-level semantics is challenging, so we implemented a pairwise comparison methodology utilizing a powerful reasoning LLM judge like [4]. Specifically, we employed Deepseek-R1 [20] to evaluate AcademicRAG's responses against those of the baseline models. The details for LLM judge is described in Appendix A.1. The evaluation employed four critical dimensions:

1. **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question?

2. **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question?

3. **Empowerment**: How well does the answer help the reader understand and make informed judgements about the topic?

4. **Overall**: This dimension assesses the cumulative performance across the three preceding criteria to identify the best overall answer.

This approach calculates robust final scores based on averaged winning percentages. The LLM evaluates both answers across the first three dimensions, determines preferences for each criterion, and then synthesizes results to identify the overall superior response.

Specifically, for each domain, we constructed 125 question-answer pairs generated by two distinct RAG frameworks. Using an LLM as a judge, we evaluated each pair across all dimensions to determine winning cases. Finally, we derive the win rates for two frameworks. To ensure fairness and mitigate positional bias, we alternated answer placement and computed final average win rates based on three trials. This multi-faceted evaluation provides nuanced understanding of each framework's strengths and limitations in academic information retrieval contexts.

3.5.2 Evaluation Results of AcademicRAG Framework

Table 3.1 presents a comparison of our AcademicRAG framework with other baseline frameworks in the agriculture and computer science (CS) domains. As shown, our method consistently outperforms nearly all other state-of-the-art (SOTA) frameworks across all dimensions in both domains. From these results, we can draw the following conclusions:

Table 3.1: AcademicRAG Win Rates (%) compared with other frameworks

Metrics	Framework	Win Rate	
		Agriculture	CS
Comprehensiveness	vs. NaiveRAG	62.8%↑	67.1%↑
	vs. LightRAG	50.0%	56.2%↑
	vs. GraphRAG	52.4%↑	51.2%↑
Diversity	vs. NaiveRAG	58.0%↑	83.2%↑
	vs. LightRAG	55.2%↑	57.8%↑
	vs. GraphRAG	50.8%↑	46.8%↓
Empowerment	vs. NaiveRAG	51.6%↑	75.7%↑
	vs. LightRAG	58.4%↑	59.8%↑
	vs. GraphRAG	54.0%↑	55.2%↑
Overall	vs. NaiveRAG	57.2%↑	77.5%↑
	vs. LightRAG	54.8%↑	56.6%↑
	vs. GraphRAG	52.4%↑	53.6%↑

Note: ↑ indicates AcademicRAG outperforming the compared framework, ↓ indicates AcademicRAG underperforming. Values are AcademicRAG win rates.

- **Wide applicability of AcademicRAG:** This consistent advantage highlights the robustness and adaptability of AcademicRAG. In agriculture, where domain-specific expertise and nuanced contextual understanding are essential, AcademicRAG effectively captures and processes complex information. In computer science, where the rapid pace of innovation and the wide distribution of knowledge present significant challenges, our framework demonstrates remarkable flexibility and accuracy. These results underscore the broad applicability and strength of AcademicRAG in handling diverse and dynamic knowledge domains. This strong performance across different academic domains highlights the versatility and adaptability of our framework, which indicates that AcademicRAG can provide more comprehensive, contextually relevant, and enriched responses compared to other RAG approaches.

- **Response Diversity:** AcademicRAG retrieves data in the form of entities and relationships, which inherently allows it to extract and utilize richer, more structured information compared to traditional RAG methods that rely on retrieving entire text chunks. This structured retrieval approach not only enhances contextual understanding but also improves the accuracy and relevance of the generated responses. Furthermore, by retrieving a subgraph rather than simply fetching one-hop neighbors from the KG, AcademicRAG can generate responses with greater diversity than LightRAG, as it incorporates a broader range of interconnected knowledge. However, while AcademicRAG improves upon LightRAG in this aspect, it does not have advantage compared with GraphRAG in terms of diversity. We hypothesize that this is due to excessive noise present in GraphRAG's community reports, which, despite increasing diversity, also introduce irrelevant or redundant information. This observation is further supported by the empowerment win rate, indicating that while GraphRAG retrieves a wider range of data, its responses may not always be as informative or useful as those generated by AcademicRAG.

- **Superiority over Graph-based RAG methods:** AcademicRAG demonstrates notable advantages over other Graph-based RAG frameworks, including GraphRAG and LightRAG, particularly within the evaluated computer science corpus. AcademicRAG employs subgraph retrieval and clue-guided keyword generation, enabling it to precisely locate relevant elements while also uncovering hidden relationships and contextual information within the KG. This enhanced retrieval mechanism allows AcademicRAG to extract deeper insights, leading to more informative, contextually rich, and well-structured responses within the scope of the evaluated corpus. By effectively synthesizing retrieved knowledge, AcademicRAG can help users grasp complex topics more efficiently and make well-informed decisions based on the available data.

To further validate the effectiveness of our approach, we conducted ablation studies on our subgraph retrieval and clues-guided keywords generation methods. These experiments systematically assessed the impact of each component by removing one while maintaining the other. The results, presented in Appendix A.2, demonstrate that both components contribute significantly to the framework's overall performance, with notable performance degradation observed when either component is removed. Additionally, we performed detailed case studies comparing AcademicRAG against other frameworks through specific queries, with comprehensive analysis and response examples provided in Appendix A.3. These case studies offer qualitative insights into how our framework achieves superior performance in real-world academic information retrieval scenarios.

Table 3.2: Comparison of Indexing, Retrieval, and Incremental Text Update Phases between GraphRAG and Our Framework

Stage	Metric	GraphRAG	Ours
Indexing	**Tokens**	$n_{\text{community}} \times 5,000 + T_{\text{extract}}$	T_{extract}
	API Calls	$(n_{\text{community}} \times 5000/C_{\text{max}}) + C_{\text{extract}}$	C_{extract}
Retrieval	**Tokens**	$610 \times 1,000$	< 500
	API Calls	$(610 \times 1,000/c_{\text{max}})$	1
Incremental	**Tokens**	$n_{\text{community}} \times 5,000 + T^{\text{new}}_{\text{extract}}$	$T^{\text{new}}_{\text{extract}}$
Text Update	**API Calls**	$(n_{\text{community}} \times 5000/C_{\text{max}}) + C^{\text{new}}_{\text{extract}}$	$C^{\text{new}}_{\text{extract}}$

3.5.3 Computing Resources Analysis

Compared to GraphRAG [4], one of the most significant advantages of AcademicRAG is its efficient utilization of computing resources. To quantify this efficiency, we compare the two frameworks based on two key metrics: the number of tokens consumed and the number of API calls required.

First, we analyze token consumption and API calls during the indexing and retrieval processes. Second, we examine how each framework handles data updates in dynamic environments, focusing on their adaptability and computational overhead. The results of this evaluation, conducted on the legal dataset, are presented in Table 3.2. In this context:

- $n_{\text{community}}$ is the number of detected community during indexing time.
- T_{extract} represents the token overhead for entity and relationship extraction.
- C_{max} denotes the maximum number of tokens allowed per API call.
- C_{extract} indicates the number of API calls required for extraction.

During indexing, GraphRAG generates community reports for $n_{\text{community}}$ detected communities in addition to extracting entities and relationships. Each community report consumes approximately 5,000 tokens, and in our experiment, about 7,000 communities were identified within the KG. This means GraphRAG incurs an additional overhead of over 35 million tokens and 1,000 API calls per domain compared to AcademicRAG.

In the retrieval process, GraphRAG retrieves around 610 community reports per query, with each report containing 1,000 tokens. In contrast, AcademicRAG optimizes retrieval, consuming fewer than 500 tokens to extract keywords instead of processing entire community reports. This efficiency is achieved through our dual-storage architecture, which seamlessly integrates graph structures and vector-based representations, eliminating the need to process lots of community reports while maintaining retrieval accuracy.

For incremental updates, both GraphRAG and AcademicRAG require $T^{\text{new}}_{\text{extract}}$ tokens for extracting newly added entities and relationships. However, GraphRAG suffers from a major inefficiency—it must delete outdated community structures and regenerate all community reports when handling new data. This leads to significant computational overhead and poor scalability in dynamic environments. In contrast, AcademicRAG efficiently integrates newly extracted entities and relationships into the existing graph without requiring any reconstruction. Our method ensures minimal overhead during both indexing and incremental updates, making AcademicRAG a more cost-effective and scalable solution for real-world applications.

3.6 Summary

The AcademicRAG framework integrates KG structures with retrieval-augmented generation to enhance academic information discovery. Its architecture comprises three key components: Graph-based Text Indexing, which transforms academic content into structured knowledge representations stored in dual graph and vector databases; Graph-Guided Query Retrieval, which leverages clue-guided keyword generation to extract both local subgraphs and global relationship networks; and Graph-Enhanced Generation, which synthesizes this information into contextually rich responses.

Evaluation results demonstrate AcademicRAG's effectiveness across two typical academic domains, consistently outperforming baseline frameworks in comprehensiveness, diversity, and user empowerment. The framework's subgraph retrieval mechanism and clue-guided keyword generation are particularly significant, enabling the system to identify complex relationships that more basic retrieval methods often miss.

AcademicRAG also achieves noteworthy efficiency in computational resource utilization, especially during retrieval and incremental updates. Unlike systems that require regenerating entire knowledge structures when new information is added, AcademicRAG's architecture allows for seamless integration of new entities and relationships, making it both effective and scalable for dynamic academic environments.

The framework provides a robust foundation for diverse downstream academic applications, including course discovery systems and research literature assistants. By capturing complex academic relationships while remaining domain-adaptable, AcademicRAG enables the development of specialized tools that can navigate intricate educational pathways, uncover cross-disciplinary connections, and personalize information discovery. This versatility makes the framework particularly valuable in addressing domain-specific challenges while maintaining consistent performance across varying academic contexts.

Chapter 4

Data Pipeline for AcademicRAG

In the AcademicRAG framework, the data pipeline is a key component to ensure the efficient operation of the system and the accurate generation of KGs. In order to provide more accurate semantic understanding and retrieval capabilities in a complex academic environment, AcademicRAG requires an automated and scalable data processing mechanism. By designing a complete end-to-end data pipeline, the system can extract structured knowledge from original documents and store it in a variety of databases, ultimately providing a strong knowledge foundation for semantic retrieval and question-answer generation.

Our data pipeline design is inspired by and references Microsoft's GraphRAG [4] and LightRAG [5]. The data pipeline is a full-process automated processing workflow from document to database. Unlike GraphRAG and LightRAG, which only support text input, our data pipeline supports multiple input formats (such as PDF, HTML, JSON, TXT), and gradually converts documents into structured KGs through a series of modular workflows. Furthermore, because we removed the community structure feature from GraphRAG, our data pipeline supports incremental input without requiring a complete reconstruction of the storage system. The data pipeline integrates functions such as document processing, index generation, and entity-relationship extraction. It ensures efficient data storage and retrieval by combining the graph database, vector database, and key-value database.

As shown in figure 4.1, AcademicRAG's data pipeline consists of three main sublevel workflows that connect to each other and are processed in a hierarchical relationship:

- **Document Process Flow** is the first layer of the pipeline. It is responsible for processing and managing the raw documents input.

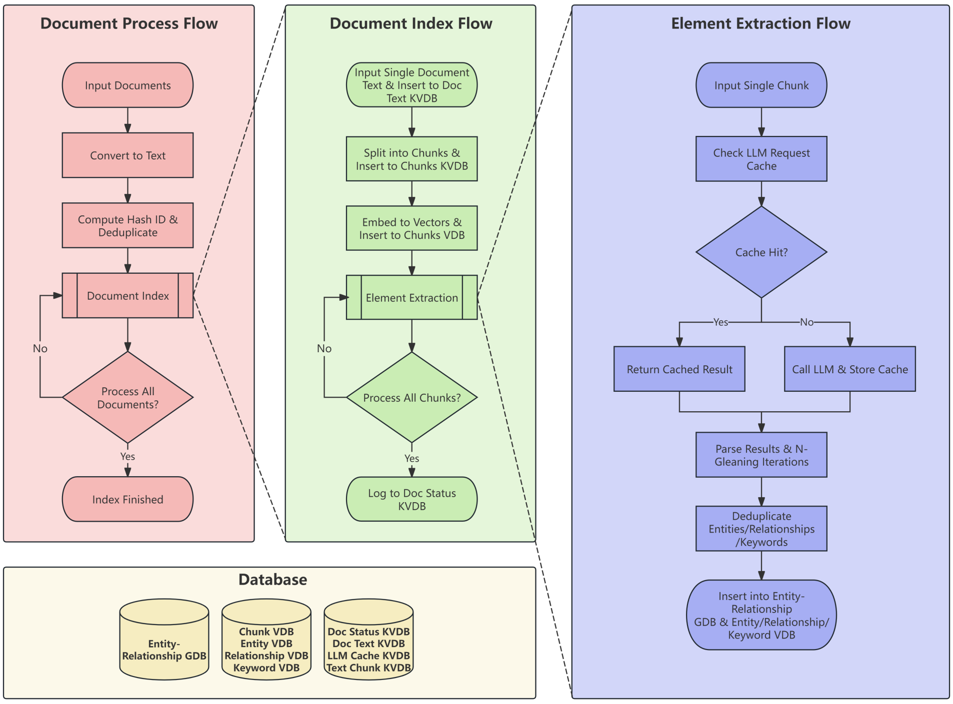

Figure 4.1: AcademicRAG Data Pipeline

Documents are controlled under a unified management framework to be passed to the **Document Index Flow** in an orderly manner, ensuring the robustness of the entire pipeline at the file management level.

- **Document Index Flow** is the second layer of the pipeline, which focuses on single document processing for the subsequent chunk indexing. This stage ensures that individual documents are processed in a structured manner while maintaining semantic information.
- **Element Extraction Flow** is the third layer of the pipeline and the most critical step, which is responsible for processing a single chunk. At this stage, entities, relationships and keywords are extracted from the chunk and stored in both graph and vector databases. It forms the basis for the query process.

The following sections provide detailed discussions on the design principles and implementation methods for each of these workflows, highlighting how AcademicRAG leverages its efficient data pipeline to construct an advanced semantic retrieval system tailored for academic contexts.

4.1 Document Process Flow

In AcademicRAG's data pipeline, the input academic documents are under the control of **Document Process Flow**. Firstly, heterogeneous academic documents are converted to standard text content. Then, a unique MD5 content hash value is computed based on each document's content, serving as an indexing key in the database and avoiding duplicate input. GraphRAG's absence of file status persistence hinders users from verifying correct processing and causes significant overhead upon restarting interrupted workflows. We addressed this by introducing a file status database "Doc Status KVDB", drawing inspiration from LightRAG [5]. After extracting basic metadata, the system sets the initial state of the document (such as `PENDING`) in the "Doc Status KVDB" based on the content hash value and records its metadata in the "Doc Status KVDB". Through this state management mechanism, the system can maintain data consistency and integrity during large-scale document processing.

After the state setting is completed, the system will filter the documents in the `PENDING` state and introduce them into the **Document Index flow**. In this process, the single document will be inserted to the semantic databases. At the same time, the state management system will update the processing state to `PROCESSING` in order to avoid redundant processing of files in multi-threaded operations.

If the document is successfully processed, the system will store the segmented text chunks and the generated embedding vectors in databases such as "Text Chunks KVDB" and "Chunk VDB". Upon successful completion, the system will update the document status to `PROCESSED`. If processing fails, the system will set the status to `FAILED` and record detailed error logs for troubleshooting and recovery.

When all documents are processed, the system terminates the current document queue processing flow. At this point, successfully indexed documents can be efficiently retrieved through AcademicRAG's query flows. Through strict state tracking and fault-tolerant design, this process ensures efficient processing of large-scale academic materials.

4.2 Document Index Flow

The **Document Index Flow** is responsible for converting input documents into structured semantic chunks for subsequent knowledge extraction. The process first receives the single document to be indexed, then it is split into semantic chunks, which serve as the basic units for **Element Extraction Flow** to facilitate refined knowledge extraction and vectorization.

After the documents are chunked, the system generates embeddings for each semantic chunk, which converts text into vector representations through embedding models. These vectors can capture the semantic information of the document content and are stored in the ”Chunk VDB” vector database. Then, the system calls the **Element Extraction Flow** to perform in-depth indexing of the content in each semantic chunk.

After element extractions are completed, the system will update the document status information to `PROCESSED` in ”Doc Status KVDB”. If a problem occurs during processing, the system will set the document status to `FAILED` and record the error log in detail for troubleshooting and recovery. Through strict state management and exception handling mechanisms, ensure that the document indexing process can be completed efficiently and reliably.

4.3 Element Extraction Flow

The **Element Extraction Flow** is the third layer and responsible for extracting entities, relationships and keywords from semantic chunks of document to build the underlying vector and graph database for AcademicRAG. This process receives single-chunked text from the **Document Index Flow** as input and calls an LLM to identify the semantic elements.

In order to reduce the consumption of computational resources, we introduced LLM cache storage like LightRAG [5]. Firstly, the system checks the LLM cache to determine whether there are extraction results for the same text chunk that have been processed before. If the cache hits, the system directly returns the cached results to avoid redundant computation. If there is no hit, the system initiates a request to the LLM to extract entities, relationships and keywords. After the first extraction, the system performs multiple rounds of gleaning optimization by parsing the results from the LLM to further improve the accuracy and completeness of the elements extraction.

After the extraction of elements is completed, the system will deduplicate the results. It ensures the uniqueness of the extracted entities, relationships and keywords. The extracted entities and relationships then are inserted into the pre-defined ”Entity-Relationship GDB” graph database to ensure efficient data storage and fast query. In addition, the system will generate vector embeddings for each entity, relationship, and keyword, storing them in the ”Entities VDB”, ”Relationship VDB”, and ”Keyword VDB” vector databases to provide rich semantic information for subsequent semantic retrieval and query.

4.4 Data Pipeline Evaluation

4.4.1 Experiment Settings

In order to evaluate the effectiveness of the AcademicRAG data pipeline in processing various types of academic content, we designed a comprehensive set of experimental methods to analyse its performance. Since the core question of the study is how to build a data pipeline that can process diverse academic content while preserving semantic relationships, we chose two types of representative academic texts as input: research articles and the course syllabuses.

For research papers, we selected 3 articles from the Machine Learning and Acoustics areas, which have some citation relationship with each other. These papers are on average 6 pages (about 7000 tokens) in length and contain complex concepts, formulas, and citation structures. For course syllabuses, we collected 5 course syllabuses for acoustics and its fundamentals from KTH Royal Institute of Technology, covering multiple levels from entry-level to advanced graduate programs.

The expected output of the AcademicRAG data pipeline is a series of structured databases, the core of which is the KG database, which is automatically generated from a given document processed through the pipeline by means of an LLM. Our evaluation therefore focuses on analyzing the quality of the KGs generated from these two types of academic texts.

Considering the shortage of widely applicable standardised assessment frameworks for KG evaluation, we adopted a qualitative approach to assess the quality of KGs generated from these two types of academic texts. Our qualitative methodology includes the following aspects:

- KG visualisation and analysis: The generated KGs are visualised to intuitively observe the structural features, connectivity and conceptual hierarchical relationships of the graphs.
- Semantic structure assessment: Assess the accuracy and completeness of the semantic relationships in the KG by means of expert review, with special attention to whether the associations between domain-specific concepts are correctly captured.

- Cross-document association analysis: Examine how the data pipeline establishes conceptual connections between different documents and assess its ability to integrate dispersed knowledge.
- Case study: Select a specific academic concept, track its representation in the KG, and analyze how the data pipeline extracts and constructs relevant knowledge structures for the concept from the raw text.

Finally, we comparatively analyze the differences in structure and content between the KGs generated from research articles and course syllabuses, assessing the adaptability of the data pipeline to handle different text formats.

Through this qualitative analysis approach, we are able to comprehensively assess the ability of the AcademicRAG data pipeline to process different academic texts, especially its performance in preserving semantic relationships. The experimental results will provide us with insights on how to design flexible data pipelines to process diverse academic content, therefore answering the second research question. In addition, the results of the evaluation of these two different academic texts will also directly inform our practical application scenarios. The KG evaluation of research paper generation will guide us to optimise the academic literature assistant application, especially in handling complex citation relationships and cross-domain conceptual connections; while the KG analysis of course syllabus generation will directly influence the design of course discovery system, helping us to improve the prerequisite course recommendation algorithms and learning path planning functions.

4.4.2 Knowledge Graph Evaluation for Research Articles

In order to systematically assess the ability of the pipeline to handle unstructured academic data, academic papers are selected as the evaluation object in this study. Although academic papers have some fixed structural elements such as abstracts and references, these structures are different between papers. Furthermore, the whole paper is often regarded as a continuous text rather than a structured segmentation from the text processing perspective of the LLMs. Therefore, it is reasonable to treat academic papers as unstructured data. The long length of the academic papers is also a significant test of the pipeline's ability to capture complex relationships in a holistic way.

In terms of specific sample selection, we adopt three papers with citation links in the field of machine learning and acoustics to construct the evaluation dataset: The landmark “Attention Is All You Need [14]” (P1), which proposes the Transformer architecture and serves as a basic reference paper in this study. The application paper in the field of acoustics, ”Transformer Transducer [41]” (P2), which directly cites paper P1. Another acoustics paper, ”Transformer-based Streaming ASR with Cumulative Attention [42]” (P3), cites both P1 and P2. This progressive citation relationship provides a multi-level testing scenario for the evaluation process to deal with complex academic associations.

In order to effectively match the textual features of academic papers and accurately evaluate the performance of the pipeline, the following settings are carried out during the experiment: (1) In view of the semantic characteristics of academic papers, the key entity categories to be extracted are clearly defined, including “Method”, “Concept”, “Paper”, “Author” and “Model”, etc., in order to ensure that the extraction task is in line with the semantic structure of academic literature. (2) Considering that academic papers are usually long in length, this study adopts a larger chunk size to ensure that the LLMs can process the complete context of the whole paper at one time. To alleviate the problem of information extraction performance degradation that may be brought about by increasing the chunk size, we increased the number of iterations for extraction, which improves the accuracy and completeness of entity recognition. These settings are designed to ensure the accuracy of information extraction to eliminate the influence of other factors in the experiment.

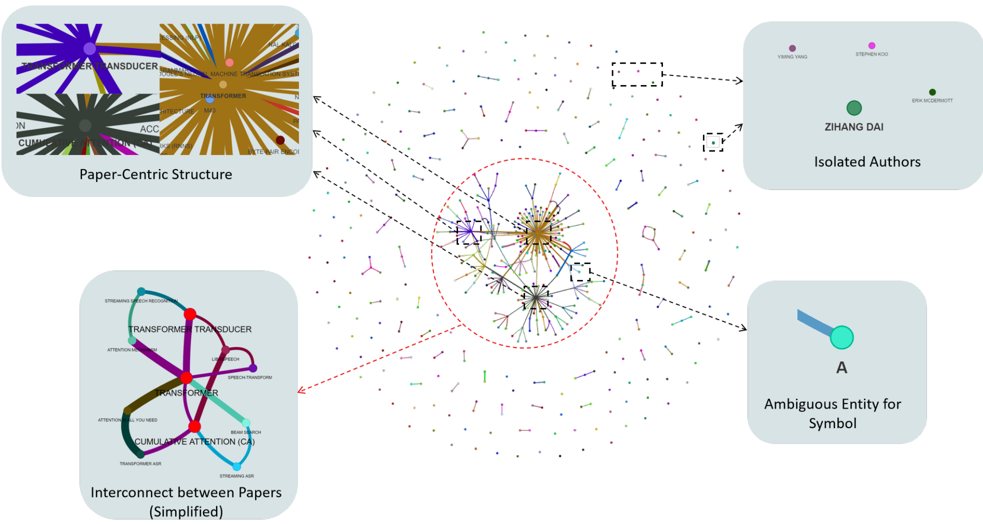

Figure 4.2: KG Visualization and Analysis for Academic Articles

Figure 4.2 shows the generated KG and some areas of interest. The KG successfully realizes a structured paper-centered representation, which completely covers the core content of the three target documents (P1, P2, and P3). One can see that the graph presents clear clustering features, in which the key elements of each of the three papers form core nodes and are interconnected through semantic associations. It is worth noting that P1 (”TRANSFORMER”), which is the base paper, exhibits significant pivotal characteristics, and its node connection density is significantly higher than that of P2 and P3. This phenomenon is in line with our expectation because the nodes in P2 (“TRANSFORMER TRANSDUCER”) and P3 (”CUMULATIVE ATTENTION”) involving the Transformer architecture are associated with P1, confirming the fundamental role of P1 in this research area.

Considering the details in the KG, we have the following findings:

- Links between Papers: The simplified subgraph in the lower left corner of Figure 4.2 shows that the system has successfully captured the deep connections between the three papers. The core nodes “TRANSFORMER”, “TRANSFORMER TRANSDUCER” and “CUMULATIVE ATTENTION” are directly connected through the shared Transformer architecture. Moreover, the system also identifies potential implicit associations, e.g., both P2 and P3 are experimented with the LibriSpeech dataset, and this common feature creates additional connections in the graph, demonstrating the framework’s ability to discover potential associations.

- Isolated Entity Problem: There are a number of isolated nodes in the graph (e.g., “ZIHANG DAI” labeled in the figure), which mainly originate from the reference section. When the entities of cited papers are split and categorized into other clusters, their corresponding author nodes are isolated due to the lack of a title node.

- Ambiguous Mathematical Symbol: Semantic disambiguation of formula symbols (e.g., N, A, etc.) remains challenging. Since the same symbols may refer to different concepts in different papers, this leads to entity confusion and introduces noise. Such ambiguous symbols can harm the performance when generating answers.

4.4.3 Knowledge Graph Evaluation for Course Syllabus

The selection of data sources for course syllabuses is an important part of validating the AcademicRAG data pipeline's ability to process semi-structured academic texts. Five acoustics and its foundation courses from the KTH Royal Institute of Technology (Sweden) were selected for evaluation. These courses represent a clear knowledge pathway as shown in Table 4.1.

Table 4.1: Selected KTH Courses for Knowledge Graph Evaluation

Course Level	Course Code and Name
Foundation Courses	SF1624 Algebra and Geometry SF1625 Calculus in One Variable
Basic Acoustics Course	SK1120 Waves
Advanced Acoustics Courses	DT1175 Sound DT2213 Musical Communication and Music Technology

There are clear knowledge dependencies and lines of learning between these courses, which are clearly noted in the course syllabuses, making them well suited for KG construction and assessment. It is worth noting that these course syllabuses are in HTML format and the information is organised in a more dispersed and fragmented manner, which provides a good test scenario for the knowledge extraction capability of the data pipeline.

When initialising the data pipeline, we configured it specifically for the structural characteristics of the course syllabus. First, multiple entity types were defined, including 'Course', 'Knowledge Concept', 'Skill', 'Person', 'Assessment', "Material", 'Learning Tool', and 'Grading scale'. Second, we increased the chunk size to ensure that each course outline is not overly segmented and that each chunk contains the course name information at the beginning of the outline. At the same time, we increased the gleaning parameter appropriately to improve the completeness of information extraction.

The results of the KG visualisation presented in the figure 4.3. Firstly, the KG exhibits a clear star-shaped structure centered on the course. Specifically, there are dense connections between the course entity and multiple related entities. This reflects the central role of the course in the KG. It demonstrates well-structured knowledge representation in graph databases.

However, the visualization presented in the figure 4.3 also reveal several key challenges for data pipelines when dealing with course syllabuses:

- Inconsistent entity representations: The figure illustrates the problem

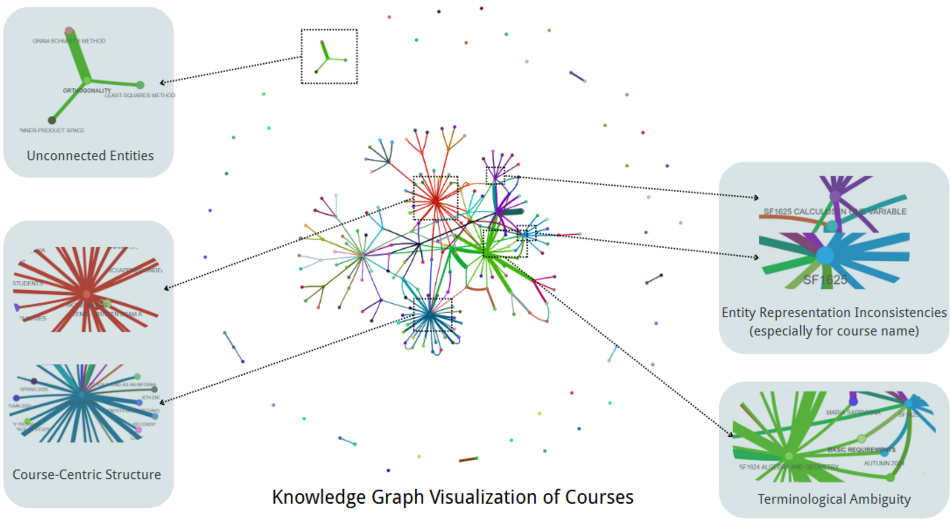

Figure 4.3: Knowledge Graph Visualization and Analysis for Course Syllabus

that the same course (e.g. SF1625) may be represented differently in different contexts. Sometimes it contains the full name and sometimes only the course code, which makes entity association more difficult.

- Terminology ambiguity: The data pipeline is ambiguous when dealing with terms such as 'BASIC REQUIREMENTS'. It can be interpreted as course knowledge requirements or prerequisites, which affects the accuracy of the relationship.
- Unconnected entities: The figure shows isolated clusters of entities such as 'GRAM-SCHMIDT METHOD', 'ORTHOGONALITY' and 'LEAST-SQUARES METHOD'. They are conceptually related but not connected to the main curriculum network.

Despite these challenges, the data pipeline successfully extracted key knowledge structures from course syllabuses. To illustrate this achievement, we merged entities representing the same course but with different names, (such as "SF1625" and "SF1625 CALCULUS IN ONE VARIABLE") into unified representations. The resulting KG visualization is shown in the figure. We highlight relationships containing terms like "prerequisite," "required," and "fundamental" in red. This visualization clearly reveals the prerequisite chain-the sequential relationships between well-defined connections. A clear representation of prerequisite relationships is essential for constructing a well-

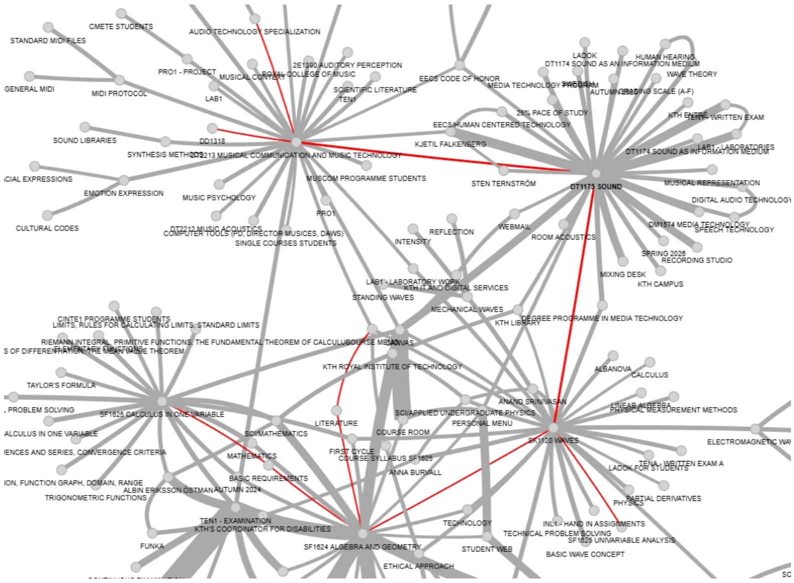

Figure 4.4: Visualization of Course Prerequisite Chain

structured course KG database. Such a database can effectively support educational pathway planning and course recommendation systems.

4.5 Summary

The AcademicRAG data pipeline implements a hierarchical, three-tier architecture designed to transform heterogeneous academic content into structured knowledge representations. Its principal components—Document Process Flow, Document Index Flow, and Element Extraction Flow—work in concert to process raw documents, segment them into semantic chunks, and extract entities, relationships, and keywords utilizing LLMs. This architecture enables fault-tolerant processing through comprehensive state management, prevents duplicate entries via content hashing, and minimizes computational overhead through strategic LLM caching mechanisms.

Experimental evaluations conducted on diverse academic corpora—including research articles with citation networks and course syllabi with prerequisite relationships—demonstrated the pipeline's capacity to construct semantically rich KGs that preserve domain-specific relationships. While successfully capturing complex interconnections between papers and course learning pathways, the system exhibited certain limitations, including entity naming inconsistencies, isolated reference nodes, and occasional terminological ambiguities that impacted graph connectivity.

The pipeline's most significant contribution lies in its support for incremental knowledge integration without requiring complete database reconstruction—a substantial improvement over previous graph-based frameworks [4] that necessitated comprehensive regeneration of community structures. This capability, coupled with its adaptability to multiple document formats and semantic extraction patterns, establishes a robust foundation for downstream applications such as course discovery systems and research literature assistants, while maintaining computational efficiency through targeted retrieval strategies rather than exhaustive community report generation.

Chapter 5

Application: Research Literature Assistant

5.1 Background

To validate the performance and adaptability of our AcademicRAG framework in real-world academic scenarios, we developed a Research Literature Assistant based on this framework. This assistant primarily focuses on two key scenarios: in-depth paper reading and literature review.

In today's research environment, many students and researchers still rely on manual methods when investigating a specific field. They often begin by screening numerous papers to identify those most relevant to their research interests. This process requires substantial time investment as they must read abstracts and introductions to determine whether papers meet their requirements. This approach not only forces them to read much unnecessary content but may also lead to overlooking crucial papers or missing potential connections due to information overload. While tools like Pasa [43] and Scite [44] can assist during this initial screening stage, they offer limited support for the subsequent in-depth paper reading process. ChatGPT's Deep Research [45] provides automated topic investigation capabilities; however, it frequently references web sources rather than scholarly literature and occasionally generates citations that cannot be verified in the academic record. Additionally, when researchers need to conduct in-depth reading of selected papers, they frequently struggle with comparing methodological advantages across different publications, understanding connections between multiple articles, identifying research trends, recognizing research gaps, and determining how to combine strengths from various approaches.

To address these challenges, we developed this Research Literature Assistant to help users efficiently and accurately navigate through academic literature. The assistant leverages our AcademicRAG framework's capabilities to provide structured analysis of research papers, enabling users to quickly identify relevant literature, understand relationships between publications, and gain deeper insights into their field of interest without the cognitive burden of processing large volumes of text manually.

5.2 Adaptation of AcademicRAG Framework

Due to the versatile and adaptable nature of our AcademicRAG framework, we were able to apply it to this downstream task with minimal modifications. For the Research Literature Assistant application scenario, we implemented the following customized adaptations:

- First, we modified the framework's prompts to specifically address the unique characteristics of academic papers, enhancing the system's ability to process scholarly content.
- Second, we added specialized functions to recognize paper titles and the introduction section, which typically contains valuable content about the publication.
- Third, we redesigned the chunking strategy to better accommodate the structural patterns common in research papers, allowing for more effective information extraction and relationship building.

These customized adaptations were designed based on our understanding of the literature review and in-depth reading scenarios. However, it is important to note that the flexibility of the AcademicRAG framework allows users to implement their own improvements according to their specific requirements. The framework's modular architecture enables researchers and developers to further refine or extend its capabilities without requiring extensive modifications to the core system, demonstrating its versatility across different academic applications. The details of these three adaptations will be described in the following sub-sections.

5.2.1 Prompts for Literature

During our evaluation of the data pipeline, we identified significant challenges when processing research papers. Specifically, the pipeline tended to generate numerous isolated entities and ambiguous entity names, primarily consisting of author names and mathematical symbols from equations. These disconnected entities reduced the effectiveness of the KG by creating fragmented information structures without proper contextual relationships.

To address this issue, we enhanced the prompt engineering for the entity extraction process. We incorporated domain-specific examples that illustrate how entities should be properly identified and connected within the context of academic papers. These examples provided the LLMs with clear patterns for recognizing author affiliations, distinguishing between mathematical symbols that represent different concepts across papers, and establishing appropriate relationships between entities.

This modification significantly improved the system's ability to correctly interpret and contextualize entities found in research papers. By providing the LLMs with a better understanding of the academic domain, we enabled the creation of more coherent and meaningful connections between entities. This enhancement made the KG more complete, effectively improving both the accuracy and comprehensiveness of graph retrieval.

5.2.2 Title and Introduction Extraction

Due to the context window limitations of LLMs, research papers typically cannot be fed in their entirety to LLMs for entity extraction. Furthermore, research by Edge et al. [4] has demonstrated that using smaller chunk sizes leads to more comprehensive extraction of entity relationships. However, when papers are divided into chunks, LLMs lose the contextual information about which paper each chunk belongs to. Therefore, extracting the paper title and appending it to each chunk becomes crucial for maintaining context.

We implemented the following approach to extract paper titles effectively:

1. Extract metadata from each text block within the first two pages of the paper.

2. Identify the maximum font size m among these text blocks.

3. Locate the first group of consecutive text blocks that meet three criteria: font size equal to m, horizontal text orientation, and bold typeface.

4. Concatenate the text from these identified blocks and obtain the paper title.

This title extraction method works effectively for the vast majority of research papers, providing essential contextual information for text chunks during processing.

When conducting literature reviews, researchers primarily focus on the initial sections of papers: the title, authors, abstract, and introduction. To adapt our application to this scenario, we developed specialized methods to extract these specific portions. Our approach follows two parallel paths depending on the paper structure. For papers with embedded tables of contents:

1. Identify and extract the embedded table of contents. Then obtain the list of top-level headings in the paper.

2. Locate the specific naming convention used for the introduction section.

3. Use keyword matching to determine its position in the paper and extract the relevant text.

For papers that lack embedded tables of contents, we employ a systematic approach to extract the introduction section. First, we identify the ”abstract” keyword, which serves as a consistent structural marker across the vast majority of scholarly publications. After successfully locating this keyword, we proceed to extract a predetermined, fixed-length portion of text that immediately follows the abstract section. This extracted content is then regarded as the introduction section of the paper.

Through this methodology, we can efficiently extract general introductory content from research papers without processing the entire document through LLMs. This approach significantly reduces computational resource requirements while still capturing the essential contextual information needed for literature review tasks.

5.2.3 Special Chunking Strategy

In the AcademicRAG framework, the standard chunking method divides input text based on a predetermined chunk size. However, this approach can result in semantic loss due to arbitrary truncation of sentences and content. During our data pipeline evaluation, we initially treated research papers as unstructured text and applied this generic chunking strategy, which led to the extraction of anomalous entities and overlooked some important connections.

Considering that research papers actually possess specific structural elements such as introductions and conclusions, we redesigned our chunking strategy to better accommodate these academic document characteristics. Our improved approach follows a hierarchical segmentation process. First, we identify the introduction and conclusion sections of the paper. Based on these markers, we divide the paper into three major segments: introduction, main body, and conclusion. We then apply the predefined chunk size parameters to each of these three segments. Importantly, during this secondary chunking process, we only create breaks at natural boundaries such as line breaks and sentence-ending periods.

This structure-aware chunking strategy significantly preserves semantic integrity and coherence within each chunk. By respecting the inherent organization of research papers and maintaining natural language boundaries, this approach minimizes information loss during the chunking process. This enhancement allows the framework to capture more accurate entity relationships and produce a more comprehensive KG representation of the academic content.

5.3 Experiment

This assistant was expected to incorporate several key capabilities essential for academic research: mapping relationships between papers, analyzing citation networks, identifying research gaps, enabling semantic search across multiple papers, and supporting topic clustering and trend analysis. These features collectively aim to provide researchers with comprehensive tools for navigating and synthesizing academic literature.

To evaluate the effectiveness of our assistant, we conducted two experimental approaches. First, we performed a literature review scenario test using a diverse dataset of approximately 80 research papers spanning multiple academic domains. In addition to the scenario test, we conducted user testing with volunteer participants to gather qualitative feedback on the system's usability and effectiveness.

5.3.1 Literature Review Scenario Test

To evaluate our Research Literature Assistant's performance, we conducted a two-phase literature review simulation using 80 research papers across multiple domains. This approach mirrors the typical researcher workflow: initial broad exploration followed by focused analysis of selected publications. For the preliminary phase, we extracted introduction sections from all papers to build an initial knowledge graph. We selected blind face restoration within computer vision as our target domain due to its methodological diversity and technical complexity. Through targeted queries, we assessed the assistant's ability to identify research trends, gaps, and relevant papers for deeper investigation. For the second phase, we input complete texts of the most relevant papers identified in previous phase to build a more comprehensive knowledge graph. We then evaluated the system through multi-dimensional questioning targeting specific capabilities: (1) **Methodological details retrieval**: Testing extraction and organization of technical information. (2) **Comparative analysis**: Evaluating synthesis of information across multiple papers. (3) **Research guidance**: Assessing actionable suggestion generation. (4) **Topic classification**: Testing categorization of papers into meaningful clusters.

The detailed query examples and system responses referenced in this section are provided in Appendix B for comprehensive review. The results show that the assistant demonstrated strong performance across all evaluation dimensions:

- **Research landscape analysis**: Table B.1 shows the assistant successfully identified major trends in blind face restoration research, including the shift from CNN-based to Transformer-based architectures, increased integration of generative priors, and growing focus on real-world applications. More significantly, it pinpointed specific research gaps such as limited 3D facial structure integration, insufficient handling of extreme degradation cases, and lack of unified evaluation standards. This comprehensive overview demonstrates the system's ability to synthesize field-wide developments from multiple sources.

- **Paper identification and contextualization**: When tasked with recommending relevant papers, the assistant not only identified appropriate publications but also provided a multi-faceted analysis of their contributions (Shown in Table B.2). For each recommended paper, the system offered insights into its methodological approach, technical innovations, and relationship to other works in the field. Furthermore, the system synthesized the collective value of the recommended papers by highlighting their contributions to multiple approaches, architectural innovations, benchmark dataset introductions, and technical advances. This comprehensive contextualization demonstrates the assistant's ability to understand the hierarchical organization of research contributions within the field.

- **Technical detail extraction**: For specific methodologies like ControlNeXt, the assistant demonstrated remarkable depth in knowledge extraction, presenting information at multiple levels of abstraction (Shown in Table B.3). It first provided a high-level overview of ControlNeXt's position as an efficient method for controllable image and video generation, then detailed its key characteristics. The system further organized technical information into logical categories covering architecture, innovations, performance characteristics, and implementation details. This detailed technical information typically requires careful reading of entire papers, highlighting the system's ability to extract and organize complex concepts.

- **Cross-paper synthesis**: When comparing two types of methods, the assistant demonstrated sophisticated analytical capabilities by structuring the comparison across multiple dimensions, including fundamental approaches, architectural differences, and performance characteristics (Shown in B.4). This structured comparative analysis required synthesizing information from multiple papers into a coherent analytical framework, demonstrating deep understanding of the technical literature.

- **Research guidance**: Table B.5 shows the assistant provided remarkably specific and actionable research guidance when asked about implementing generative priors. Rather than offering generic advice, it delivered concrete implementation strategies with specific examples from the literature. The system also highlighted practical implementation tips and warned against common implementation pitfalls, demonstrating the system's capacity to support practical research planning based on comprehensive literature analysis.

- **Methodological classification**: One can see the assistant organized the literature into coherent categories from Table B.6. Beyond simple categorization, it provided detailed explanations of each category's defining characteristics and representative works. This detailed categorization extended across multiple methodological approaches, including model distillation, domain-specific applications, foundational works, and hybrid approaches. This feature can help researchers understand the methodological landscape and relationships between different approaches.

These results confirm that our Research Literature Assistant successfully fulfills its intended functions: mapping paper relationships, identifying research gaps, enabling semantic search across papers, and supporting topic clustering and trend analysis.

5.3.2 User Test

User testing provided valuable insights into the real-world performance of our Research Literature Assistant, helping us develop a more nuanced understanding of its strengths and limitations in practical applications. For this evaluation, two graduate students with machine learning backgrounds were invited to use the assistant freely for paper input and query tasks, while also comparing its performance with online LLM-based question-answering systems such as ChatGPT.

Recognizing the inherent limitations of online LLMs in processing multiple research papers simultaneously, we structured the comparison to focus specifically on in-depth reading scenarios involving a small number of papers. This approach ensured a feasible comparison while still testing the aspect of literature review. To establish a fair comparative framework, we deliberately configured our assistant to use a weaker model (GPT-4o-mini) against the more powerful GPT-4o model used in the online LLM. This disparity actually placed our system at a computational disadvantage, making any performance advantages more significant.

Upon completion of the testing sessions, participants were asked to provide comprehensive feedbacks regarding their experience. This qualitative assessment encompassed various aspects including response quality, comprehensiveness in information retrieval, accuracy in relationship identification, and overall user satisfaction. Participants were asked to share their observations and experiences through open-ended discussions focused on these specific aspects. This conversational approach allowed us to capture both their immediate impressions and considered reflections on the comparative advantages of our KG-enhanced approach versus the standard LLMs interaction model.

This user-centered evaluation methodology complemented our technical assessments, providing crucial perspectives on how researchers might integrate such tools into their actual workflow and identifying potential refinements to enhance the assistant's practical utility in academic research contexts.

5.3.2.1 Feedbacks from Users

The user testing yielded several insightful observations regarding the strengths and limitations of our Research Literature Assistant. Participants provided detailed feedback across multiple dimensions of the system's performance:

- Regarding the assistant's analytical capabilities, users found the topic clustering and trend analysis functionality satisfactory, noting its effectiveness in distinguishing papers with prominent features. One user said, ”Clusters without obvious semantic relationships tend to be overlooked”, which shows that the system might miss identifying relationships between papers when those connections are more subtle or less explicitly defined in the semantic structure.

- Users reported that query results were highly prompt-dependent, with significant variance in response quality based on how questions were formulated. We attribute this sensitivity to our framework's reliance on vector similarity matching for KG retrieval, which can sometimes limit flexibility in query interpretation compared to pure LLMs.

- The assistant received particularly positive feedback for its research gap identification and semantic search capabilities. Users highlighted the system's ability to clearly articulate unexplored research directions and efficiently locate relevant information across multiple papers, suggesting these features provide substantial value for literature review tasks.

- Paper relationship mapping and citation network analysis functionalities were also highly regarded. Participants appreciated the assistant's ability to visualize connections between publications and trace the evolution of ideas across the literature, facilitating deeper understanding of research lineages.

When compared with online LLMs processing of the same papers, our assistant demonstrated two significant advantages: the capacity to accommodate and synthesize information from a larger corpus of papers, and noticeably higher accuracy in responses. Users specifically valued the system's ability to retrieve highly detailed information from papers, indicating that the KG approach preserves specificity that might be lost in pure LLMs processing.

A particularly noteworthy strength was the assistant's factual reliability. Users observed that unlike online LLMs, which sometimes incorporated external knowledge that could be incorrect or irrelevant to the papers being discussed, our assistant consistently limited responses to information contained within the knowledge base, reducing the risk of hallucinations.

Despite these advantages, users identified several practical limitations. The deployment process was considered less convenient than web-based LLMs interfaces, potentially creating barriers for users without programming experience, especially for users from non-scientific backgrounds. The indexing process introduced a time delay not present in standard LLMs interactions. Additionally, the system demonstrated relatively weaker comprehension of non-textual elements such as images, formulas, and tables compared to online alternatives—a limitation we attribute to the absence of multi-modal models that could be addressed in future iterations.

These user insights provide valuable direction for ongoing development, highlighting both the substantial advantages of our KG-enhanced approach and specific areas where usability and feature enhancements could further improve the assistant's utility for academic research.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions drawn from our research on Graph-based RAG for Academic Resource Discovery, analyses the limitations of our current approach, and outlines potential directions for future work. First, we summarize the main contributions of this research with respect to our initial research questions. Next, we discuss the constraints and limitations that affected our implementation and results. Finally, we identify outstanding challenges and suggest paths for continued research to further framework improvement and application in academic contexts.

6.1 Conclusions

In this project, we developed AcademicRAG, a novel graph-based retrieval-augmented generation framework specifically tailored for academic resource discovery. Building upon existing open-source frameworks, we implemented innovative approaches to enhance both retrieval quality and computational efficiency. Our framework leverages LLMs to extract entities and relationships from input texts, constructing comprehensive KGs that preserve semantic connections inherent in academic materials. We introduced two key technical contributions: a clue-guided keywords retrieval method that reduces hallucination and improves retrieval precision, and a subgraph-based local information extraction technique that captures deeper contextual relationships compared to other approaches. These enhancements not only improved the quality of graph retrieval but also significantly reduced the computational resources required during the graph indexing phase. Evaluation on the UltraDomain dataset demonstrated that AcademicRAG consistently outperforms Microsoft's GraphRAG [4] and LightRAG [5] across multiple dimensions including comprehensiveness, diversity, and user empowerment.

Furthermore, we developed a robust data pipeline architecture to process various formats of semi-structured and unstructured academic texts. This three-tiered processing pipeline efficiently handles document transformation, semantic chunking, and entity-relationship extraction, enabling accurate KG construction while preserving critical semantic relationships both within and between documents. A key innovation in our pipeline design is the integration of multiple database technologies—combining graph databases for relationship storage, vector databases for semantic embeddings, and key-value databases for efficient document management and state tracking. This multi-database approach enables more flexible and comprehensive information retrieval while maintaining high performance. The pipeline's design emphasizes modularity and fault tolerance, allowing for seamless integration of new academic content without requiring complete database reconstruction.

Finally, we demonstrated the practical applicability of our framework by developing a research literature assistant capable of facilitating literature reviews and in-depth paper reading. This downstream application required minimal framework modifications, highlighting AcademicRAG's versatility and ease of adaptation. The assistant performed exceptionally well in both simulated scenario testing and real-user evaluations, providing contextually relevant information and meaningful insights from academic resources. This successful implementation validates our framework's design principles and confirms its value for enhancing academic resource discovery across various domains.

Through our research, we have addressed our initial research questions. Regarding RQ1 (How can graph databases and language models be effectively integrated in a Graph-based RAG architecture to enable efficient academic resource discovery?), we found that the optimal integration involves using LLMs for intelligent entity, relationship and content keywords extraction, coupled with a dual retrieval strategy that combines subgraph traversal and context-guided keyword generation. This approach preserves the semantic richness of academic texts while enabling efficient retrieval without the computational overhead of traditional community-based methods. The elimination of community reports and introduction of content keywords significantly reduced token consumption during both indexing and retrieval phases.

For RQ2 (What architectural patterns enable a flexible data pipeline capable of processing both unstructured and semi-structured academic content while preserving semantic relationships?), our research demonstrated that a hierarchical, modular pipeline architecture with distinct processing flows—document processing, document indexing, and element extraction—provides the necessary flexibility and robustness. The integration of multiple specialized databases (graph, vector, and key-value) creates a comprehensive storage system that maintains both structural relationships and semantic meanings. Critical to this architecture is the state management mechanism, which ensures processing integrity across large-scale document collections and enables fault-tolerant, incremental updates without requiring complete database reconstruction.

Addressing RQ3 (How can the Graph-based RAG framework be effectively applied to downstream academic tasks, and what is its performance in real-world applications?), our implementation of a research literature assistant demonstrated that the AcademicRAG framework can be effectively adapted to specific academic use cases with minimal modifications. The key to successful application lies in domain-specific prompt engineering and an appropriate chunking strategy to match the task requirements. Our research literature assistant showed exceptional performance in real-world scenarios, providing contextually relevant insights from academic papers and enabling users to explore complex research relationships efficiently. User evaluations indicated high satisfaction with the assistant's ability to identify cross-paper connections and synthesize information from multiple sources. The successful deployment of this application validates the framework's versatility and confirms its practical value in enhancing academic resource discovery and utilization in real-world settings.

6.2 Limitations

While our AcademicRAG framework demonstrated significant advantages in academic resource discovery, several limitations were identified during development and evaluation:

- The framework's indexing process for complex academic texts often requires multiple rounds of entity extraction. This iterative refinement, while improving extraction quality, leads to increased computational resource consumption. In academic contexts with intricate concept relationships and specialized terminology, multiple extraction passes are sometimes necessary to capture the full semantic richness of the text. This requirement creates a trade-off between extraction quality and computational efficiency that must be carefully managed, particularly when processing large document collections.

- Our data pipeline exhibits certain processing limitations when handling diverse academic content. We observed instances of relationship omissions, ambiguous entity naming, and isolated entities disconnected from the main KG. While the pipeline successfully preserves most semantic relationships and functions effectively for the majority of text inputs, these issues can impact retrieval quality in specific contexts. For example, when processing course syllabi, we found inconsistent entity representations where the same course might appear with different naming patterns across documents. Similarly, in research papers, ambiguous mathematical symbols sometimes created entity confusion. Although these limitations can be addressed through domain-specific customization, they represent inherent challenges in the general-purpose pipeline design.
- Feedback from application deployment revealed practical implementation barriers. Users reported that the framework installation process presented a moderate technical challenge, requiring familiarity with graph databases and language model configuration. Additionally, the KG construction phase requires significant processing time, especially for large document collections, creating a noticeable wait period before the system becomes operational. This initial setup overhead might discourage adoption in time-sensitive academic environments where immediate results are expected.
- The current KG construction approach has limitations in identifying subtle clustering relationships (e.g. some semantically insignificant clusters). While our framework effectively captures explicit connections between entities, it struggles to detect implicit or less obvious clustering patterns that exist in academic knowledge structures. The system can readily identify and represent clearly defined relationships, but more nuanced associations—particularly those requiring domain expertise to recognize—often remain undetected. This limitation restricts the framework's ability to uncover hidden knowledge patterns that might be valuable for comprehensive academic understanding, especially in interdisciplinary areas where connections between concepts are not immediately apparent.

These limitations highlight important areas for future refinement of the

AcademicRAG framework. Despite these constraints, the system remains effective for its intended purpose, providing enhanced academic resource discovery capabilities with demonstrable advantages over existing approaches.

6.3 Future work

Building upon the findings and limitations of our current research, several promising directions for future work have been identified. These avenues of exploration could further enhance the capabilities and applications of our AcademicRAG framework.

6.3.1 Multimodal Graph-based RAG Framework

Our current framework is limited to processing textual information, which represents only one dimension of academic content. Academic materials frequently include various media formats such as images, diagrams, videos, and audio recordings that contain valuable information not captured in text alone.

A significant extension would be developing a multimodal Graph-based RAG framework capable of processing and integrating information from diverse media types. This would involve:

- Implement multi-modal models to extract entities and relationships from non-textual content, such as recognizing diagrams in research papers, identifying visual concepts in educational videos, or extracting structured information from tables and charts.

- Design cross-modal KG structures that can represent relationships between entities from different modalities (e.g., connecting textual descriptions with their corresponding visual representations).

- Develop retrieval mechanisms that can effectively search across modalities, enabling queries to retrieve relevant information regardless of the source format.

Such a multimodal framework would provide a more comprehensive approach to academic knowledge representation and discovery, better reflecting the diverse nature of academic resources and enhancing the system's ability to support complex information needs.

6.3.2 Exploration with Advanced LLMs

Our current implementation utilizes LLMs with parameters under 100 billion, which may impose certain performance limitations. As more advanced models become available, exploring their integration with our framework represents a promising research direction. Future work could investigate:

- Implement AcademicRAG with state-of-the-art models featuring larger parameter counts to evaluate potential improvements in entity extraction accuracy, relationship identification, and response generation quality.
- Analyze the trade-offs between model size, computational requirements, and performance gains to identify optimal configurations for different academic contexts.
- Explore techniques for model distillation or efficient fine-tuning that could enable smaller, specialized models to achieve comparable performance to larger models in academic domains.

This research would help determine whether the current limitations in KG construction and query performance are inherent to our framework design or are constrained by the capabilities of the underlying language models employed.

6.3.3 Enhanced Retrieval Mechanisms

While our current retrieval approach demonstrates advantages over existing methods, there remains significant potential for improvement, particularly in identifying subtle relationships within the KG. Future work on retrieval enhancement could focus on:

- Developing advanced clustering algorithms specifically designed for academic KGs that can better identify implicit or less obvious relationships between concepts.
- Implementing graph neural networks to learn domain-specific patterns and improve the retrieval of relevant subgraphs based on user queries.
- Investigating personalized retrieval mechanisms that adapt to individual users' knowledge backgrounds and information needs, potentially incorporating user interaction patterns to refine retrieval priorities over time.

These enhancements could address the current limitations in detecting subtle clustering relationships and further improve the relevance and comprehensiveness of retrieved information, thereby enhancing the overall quality of responses generated by the system.

By pursuing these research directions, future work can build upon the foundation established by AcademicRAG, potentially transforming how academic resources are discovered, accessed, and utilized across various disciplines and contexts.

6.4 Reflections

This research on the AcademicRAG framework presents meaningful implications beyond its technical contributions, particularly in economic, environmental, and social domains.

From an economic and environmental perspective, our framework significantly improves computational efficiency compared to existing approaches. Microsoft's GraphRAG implementation [4] relies on community structures that require extensive LLMs processing for generating and retrieving community reports during both indexing and query phases. By redesigning the architecture to eliminate these resource-intensive components while maintaining performance, our AcademicRAG framework substantially reduces computational requirements. This improvement translates directly to lower operational costs and decreased energy consumption, contributing to more sustainable AI development in an era of growing computational demands.

Socially, our framework enhances access to academic knowledge by making information retrieval more efficient and contextually aware, supporting broader educational goals and potentially facilitating interdisciplinary collaboration. By requiring fewer computational resources, our approach also makes advanced knowledge discovery tools more accessible to users with limited resources.

The thesis contributes to the United Nations Sustainable Development Goals, particularly SDG 4 (Quality Education) by enhancing access to educational resources, and SDG 9 (Industry, Innovation and Infrastructure) by developing sustainable technological infrastructure for knowledge management.

References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas *et al.*, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990. [Page 1.]

[2] C. D. Manning, *An introduction to information retrieval*, 2009. [Page 1.]

[3] Y. Gao, Y. Xiong, X. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey." [Online]. Available: http://arxiv.org/abs/2312.10997 [Pages x, 2, 13, and 14.]

[4] D. Edge, H. Trinh, N. Cheng *et al.*, "From local to global: A graph RAG approach to query-focused summarization." [Online]. Available: http://arxiv.org/abs/2404.16130 [Pages 2, 7, 12, 17, 19, 21, 25, 29, 30, 31, 32, 35, 38, 49, 52, 60, and 66.]

[5] Z. Guo, L. Xia, Y. Yu *et al.*, "LightRAG: Simple and fast retrieval-augmented generation." [Online]. Available: http://arxiv.org/abs/2410.05779 [Pages 2, 7, 9, 17, 19, 28, 31, 38, 40, 41, and 60.]

[6] J. Wu, J. Zhu, Y. Qi *et al.*, "Medical graph RAG: Towards safe medical large language model via graph retrieval-augmented generation." [Online]. Available: http://arxiv.org/abs/2408.04187 [Pages 2, 18, 19, 21, and 24.]

[7] H. Qian, P. Zhang, Z. Liu *et al.*, "MemoRAG: Moving towards next-gen RAG via memory-inspired knowledge discovery." [Online]. Available: http://arxiv.org/abs/2409.05591 [Pages 2, 7, and 30.]

[8] B. Peng, Y. Zhu, Y. Liu *et al.*, "Graph retrieval-augmented generation: A survey." [Online]. Available: http://arxiv.org/abs/2408.08921 [Pages 2, 15, and 21.]

[9] T. Fan, J. Wang, X. Ren, and C. Huang, "MiniRAG: Towards extremely simple retrieval-augmented generation." [Online]. Available: http://arxiv.org/abs/2501.06713 [Page 2.]

[10] Y. Hu, Z. Lei, Z. Zhang *et al.*, "GRAG: Graph retrieval-augmented generation." [Online]. Available: http://arxiv.org/abs/2405.16506 [Pages 2, 18, 21, and 28.]

[11] X. He, Y. Tian, Y. Sun *et al.*, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering." [Online]. Available: http://arxiv.org/abs/2402.07630 [Pages 2 and 21.]

[12] Z. Liu, "AcademicRAG: Knowledge graph enhanced retrieval-augmented generation for academic resource discovery." [Page 6.]

[13] P. Lewis, E. Perez, A. Piktus *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks." [Online]. Available: http://arxiv.org/abs/2005.11401 [Pages 7, 13, and 31.]

[14] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need." [Online]. Available: http://arxiv.org/abs/1706.03762 [Pages 11 and 44.]

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding." [Online]. Available: http://arxiv.org/abs/1810.04805 [Page 11.]

[16] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training." [Online]. Available: https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035 [Page 11.]

[17] T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners." [Online]. Available: http://arxiv.org/abs/2005.14165 [Page 11.]

[18] J. Wei, X. Wang, D. Schuurmans *et al.*, "Chain-of-thought prompting elicits reasoning in large language models." [Online]. Available: http://arxiv.org/abs/2201.11903 [Page 11.]

[19] OpenAI, J. Achiam, S. Adler *et al.*, "GPT-4 technical report." [Online]. Available: http://arxiv.org/abs/2303.08774 [Pages 11 and 14.]

[20] DeepSeek-AI, D. Guo, D. Yang *et al.*, "DeepSeek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning." [Online]. Available: http://arxiv.org/abs/2501.12948 [Pages 11 and 32.]

[21] S. Ji, S. Pan, E. Cambria *et al.*, "A survey on knowledge graphs: Representation, acquisition and applications," vol. 33, no. 2, pp. 494–514. doi: 10.1109/TNNLS.2021.3070843. [Online]. Available: http://arxiv.org/abs/2002.00388 [Page 11.]

[22] S. Bubeck, V. Chandrasekaran, R. Eldan *et al.*, "Sparks of artificial general intelligence: Early experiments with GPT-4." [Online]. Available: http://arxiv.org/abs/2303.12712 [Page 12.]

[23] L. Ouyang, J. Wu, X. Jiang *et al.*, "Training language models to follow instructions with human feedback." [Online]. Available: http://arxiv.org/abs/2203.02155 [Page 12.]

[24] Z. Shao, P. Wang, Q. Zhu *et al.*, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models." [Online]. Available: http://arxiv.org/abs/2402.03300 [Page 12.]

[25] H. Touvron, T. Lavril, G. Izacard *et al.*, "LLaMA: Open and efficient foundation language models." [Online]. Available: http://arxiv.org/abs/2302.13971 [Page 12.]

[26] A. Hogan, E. Blomqvist, M. Cochez *et al.*, "Knowledge graphs," vol. 54, no. 4, pp. 1–37. doi: 10.1145/3447772. [Online]. Available: http://arxiv.org/abs/2003.02320 [Page 12.]

[27] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," vol. 57, no. 10, pp. 78–85. doi: 10.1145/2629489. [Online]. Available: https://dl.acm.org/doi/10.1145/2629489 [Pages 12 and 16.]

[28] P. Chen, Y. Lu, V. W. Zheng *et al.*, "KnowEdu: A system to construct knowledge graph for education," vol. 6, pp. 31 553–31 563. doi: 10.1109/ACCESS.2018.2839607 Conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/document/8362657 [Page 12.]

[29] M. Yasunaga, A. Bosselut, H. Ren *et al.*, "Deep bidirectional language-knowledge graph pretraining." [Online]. Available: http://arxiv.org/abs/2210.09338 [Page 13.]

[30] A. Saha, V. Pahuja, M. M. Khapra *et al.*, "Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph." [Online]. Available: http://arxiv.org/abs/1801.10314 [Page 13.]

[31] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering." [Online]. Available: http://arxiv.org/abs/1809.02789 [Page 13.]

[32] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension." [Online]. Available: http://arxiv.org/abs/1705.03551 [Page 15.]

[33] N. Pipitone and G. H. Alami, "LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain." [Online]. Available: http://arxiv.org/abs/2408.10343 [Page 15.]

[34] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking retrieval-augmented generation for medicine." [Online]. Available: http://arxiv.org/abs/2402.13178 [Page 15.]

[35] N. F. Liu, K. Lin, J. Hewitt *et al.*, "Lost in the middle: How language models use long contexts." [Online]. Available: http://arxiv.org/abs/2307.03172 [Page 17.]

[36] Y. Hu, Z. Lei, Z. Dai *et al.*, "CG-RAG: Research question answering by citation graph retrieval-augmented LLMs." [Online]. Available: http://arxiv.org/abs/2501.15067 [Pages 18, 19, and 21.]

[37] R. Jia, B. Zhang, S. J. R. Méndez, and P. G. Omran, "Leveraging large language models for semantic query processing in a scholarly knowledge graph." [Online]. Available: http://arxiv.org/abs/2405.15374 [Pages 18 and 19.]

[38] M. Huang, C. Bu, Y. He, and X. Wu, "How to mitigate information loss in knowledge graphs for GraphRAG: Leveraging triple context restoration and query-driven feedback." [Online]. Available: http://arxiv.org/abs/2501.15378 [Page 21.]

[39] B. Sarmah, B. Hall, R. Rao *et al.*, "HybridRAG: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction." [Online]. Available: http://arxiv.org/abs/2408.04948 [Page 21.]

[40] Qwen, A. Yang, B. Yang *et al.*, "Qwen2.5 technical report." [Online]. Available: http://arxiv.org/abs/2412.15115 [Page 31.]

[41] Q. Zhang, H. Lu, H. Sak *et al.*, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-t loss." [Online]. Available: http://arxiv.org/abs/2002.02562 [Page 44.]

[42] M. Li, S. Zhang, C. Zorila, and R. Doddipatla, "Transformer-based streaming ASR with cumulative attention." [Online]. Available: http://arxiv.org/abs/2203.05736 [Page 44.]

[43] Y. He, G. Huang, P. Feng *et al.*, "PaSa: An LLM agent for comprehensive academic paper search." [Online]. Available: http://arxiv.org/abs/2501.10120 [Page 50.]

[44] AI for research | scite. [Online]. Available: https://scite.ai [Page 50.]

[45] Introducing deep research. [Online]. Available: https://openai.com/index/introducing-deep-research/ [Page 50.]

Appendix A

Supporting materials

The GitHub repository for our AcademicRAG framework: `https://github.com/shua-chen/academicRAG`

A.1 LLM Judge Evaluation Protocol

The following prompt was used to configure the large language model (Deepseek-R1) for conducting pairwise evaluations of responses generated by different RAG frameworks. The evaluation was structured to assess three key dimensions—**Comprehensiveness**, **Diversity**, and **Empowerment**—along with an overall judgment.

A.1.1 System Instruction

> **—Role—**
> You are an expert tasked with evaluating two answers to the same question based on three criteria: Comprehensiveness, Diversity, and Empowerment.
>
> - **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question?
> - **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question?
> - **Empowerment**: How well does the answer help the reader understand and make informed judgments about the topic?
>
> For each criterion, choose the better answer (either Answer Petter or Answer Bob) and explain why. Then, select an overall winner based on

these three categories.

A.1.2 Evaluation Template

```
Here is the question:
{query}

Here are the two answers:

**Answer Petter:**
{answer1}

**Answer Bob:**
{answer2}

Evaluate both answers using the three criteria listed
above and provide detailed reasoning for each criterion.

Output your evaluation in the following JSON format:

{
  "Comprehensiveness": {
    "Reasoning": "[Provide reasoning here]",
    "Winner": "[Answer Petter or Answer Bob]"
  },
  "Diversity": {
    "Reasoning": "[Provide reasoning here]",
    "Winner": "[Answer Petter or Answer Bob]"
  },
  "Empowerment": {
    "Reasoning": "[Provide reasoning here]",
    "Winner": "[Answer Petter or Answer Bob]"
  },
  "Overall Winner": {
    "Reasoning": "[Provide reasoning here]",
    "Winner": "[Answer Petter or Answer Bob]"
  }
}
```

A.1.3 Implementation Details

For each query-response pair evaluation, the framework names ("Petter" and "Bob") served as placeholders for the actual RAG systems being compared (AcademicRAG, GraphRAG, LightRAG, or NaiveRAG). To mitigate potential positional bias, we conducted three evaluations per pair, alternating the order of presentation, and averaged the results to determine final win rates. The structured JSON output facilitated automated aggregation of win/loss statistics across the evaluation corpus of 125 questions per domain.

A.2 Ablation studies

Table A.1: Ablation Studies on Win Rates (%) Compared with GraphRAG

Metrics	Framework	Win Rate	
		Agriculture	CS
Comprehensiveness	vs. AcademicRAG	52.4%↑	51.2%↑
	vs. -clues	53.6%↑	51.0%↑
	vs. -subgraph	50.8%↑	49.2%↓
Diversity	vs. AcademicRAG	50.8%↑	46.8%↓
	vs. -clues	48.4%↓	43.0%↓
	vs. -subgraph	51.6%↑	48.0%↓
Empowerment	vs. AcademicRAG	54.0%↑	55.2%↑
	vs. -clues	60.8%↑	54.6%↑
	vs. -subgraph	52.8%↑	59.3%↑
Overall	vs. AcademicRAG	52.4%↑	53.6%↑
	vs. -clues	57.2%↑	51.8%↑
	vs. -subgraph	53.6%↑	55.2%↑

Note: ↑ indicates the noted framework outperforming the GraphRAG, ↓ indicates underperforming. Values are win rates.

To validate the effectiveness of our proposed methods, we conduct ablation studies to assess the impact of subgraph retrieval and clue-guided keyword generation. The results, presented in Table A.1, illustrate the contributions of these components to the overall performance of the AcademicRAG framework. We analyze the effects of subgraph and clues by omitting one of them during the retrieval stage, then comparing the ablated models against GraphRAG on the evaluation dataset. Our observations for the different variants are listed as follows:

- **Impact of Clues:** The -clues model eliminates the clue-guided keyword generation process during keyword extraction. Instead, it relies solely on LLM-generated keywords, derived directly from the user's query, to retrieve relevant nodes and edges from the KG. Without the guidance of contextual clues, the LLM generates keywords based solely on its internal knowledge, which often leads to hallucinations—producing keywords that are irrelevant to the actual KG. This misalignment negatively impacts the retrieval stage, reducing the amount of retrieved information and ultimately diminishing the diversity of the generated answers.

- **Impact of Subgraph:** The -subgraph model modifies the retrieval stage by fetching one-hop neighbors instead of retrieving entire subgraph. While this approach introduces a greater number of neighboring nodes, thereby allowing the model to generate more diverse responses, it also comes with significant drawbacks. Many of the retrieved neighbor nodes are irrelevant to the query, adding unnecessary noise to the retrieval process. Additionally, some crucial interconnected nodes that could provide deeper contextual insights are overlooked, resulting in responses that are less comprehensive.

The results of our ablation studies highlight the critical role of both subgraph retrieval and clue-guided keyword generation in the AcademicRAG framework. The removal of either component leads to noticeable declines in some dimensions, demonstrating their effectiveness and necessity. Subgraph retrieval ensures that the model captures potential relationships and retrieves comprehensive contextual information, while clue-guided keyword generation prevents hallucinations by aligning keyword selection with actual KG content. These findings reinforce the importance of these mechanisms in enhancing retrieval precision, maintaining contextual relevance, and improving overall response diversity within the AcademicRAG framework.

A.3 Case Study

Beyond the advantages reflected in statistical data, to provide a more intuitive analysis of AcademicRAG's answer generation capability on academic texts, this section will present two cases, each offering a direct comparison of our framework with LightRAG and GraphRAG in actual evaluations.

The case study in Table A.2 demonstrates AcademicRAG's ability to generate structured, actionable responses through its subgraph-aware retrieval mechanism. For the query regarding hybrid models in ride-sharing segmentation, GraphRAG produces a conventional answer focusing on basic model combinations (regression + clustering) and dataset applications. While technically correct, its response exhibits three critical limitations: (1) Linear presentation of methods without hierarchical organization, (2) Over-reliance on textbook case studies rather than operational frameworks, and (3) Limited connection between technical solutions and practical implementation.

Table A.3 reveals similar limitations in LightRAG's approach, despite its graph-based foundation. LightRAG organizes its response into five discrete components—regression, clustering, combination, evaluation, and data transformation—but fails to establish meaningful connections between these elements. For instance, while it mentions performance evaluation metrics (cross-validation, WCSS), it does not specify how these metrics should be applied iteratively to optimize hybrid models. The ”Application in Ride-Sharing” section merely lists generic objectives (”segment users”, ”predict behavior”) without detailing implementation workflows. This fragmented structure stems from LightRAG's reliance on keyword-based one-hop neighbor retrieval, which captures isolated concepts (”regression”, ”clustering”) but misses higher-order relationships between methodological components and operational requirements. Consequently, LightRAG's answer resembles a textbook summary rather than an executable solution framework.

AcademicRAG overcomes these limitations through its unique retrieval architecture. The clue-guided keyword generation first identifies core concepts (”hybrid models”, ”user segmentation”, ”dynamic refinement”) and latent requirements (”implementation steps”, ”scalability considerations”). This triggers a targeted subgraph retrieval that connects regression techniques (e.g., decision trees) with advanced clustering methods (hierarchical clustering) and system design principles. The retrieved subgraph reveals four distinct hybrid approaches through entity relationships that both GraphRAG and LightRAG missed, including the hierarchical clustering-regression combination and ensemble methods.

The LLM evaluation highlights AcademicRAG's superiority across all dimensions. In comprehensiveness, the framework's ability to retrieve interconnected concepts from the subgraph allows coverage of 4 hybrid approaches versus GraphRAG's 2 and LightRAG's 3. The diversity advantage stems from synthesizing cross-domain knowledge – while baseline models focus on machine learning components, AcademicRAG incorporates software design patterns from the computer science corpus through subgraph relationships. Empowerment is achieved through operational guidance, where AcademicRAG converts retrieved knowledge into implementable workflows (e.g., explicit evaluation phase integration), directly addressing the implementation gaps observed in both GraphRAG and LightRAG responses.

Table A.3 extends our comparative analysis by examining the performance differential between LightRAG and AcademicRAG in response to the identical query on hybrid models for user segmentation in ride-sharing. The juxtaposition of these frameworks reveals notable distinctions in their respective approaches and output quality. While both systems present structured responses with enumerated methodologies, AcademicRAG demonstrates superior organizational coherence and conceptual depth. Specifically, LightRAG's response, though informative with its five-point structure covering regression models, clustering techniques, combined approaches, evaluation metrics, and data transformation, lacks the integrated theoretical framework evident in AcademicRAG's presentation. AcademicRAG's response is distinguished by its systematic progression from foundational hybrid models to more sophisticated techniques like hierarchical clustering and ensemble methods, coupled with a concrete practical example that delineates implementation steps. The LLM Decision section further validates this qualitative assessment, highlighting AcademicRAG's superiority across all evaluation dimensions—comprehensiveness, diversity, and empowerment—thereby reinforcing the framework's enhanced capability to generate academically rigorous and practically applicable responses in specialized domains such as transportation analytics and user behavior modeling.

Table A.2: Case Study: Comparison Between GraphRAG and AcademicRAG

Query	What hybrid models (e.g., regression + clustering) enhance user segmentation in ride-sharing?
GraphRAG	**Hybrid Models for User Segmentation in Ride-Sharing** User segmentation in ride-sharing platforms is critical for enhancing user experience and optimizing business strategies. Hybrid models combining regression and clustering techniques can significantly improve segmentation accuracy. **Regression Models:** Used to predict metrics like ride frequency, duration, and distance. For example, the **Linear Model** and **Decision Tree Model** were applied to the **Bike Sharing Dataset**, with the Decision Tree achieving better performance (RMSLE: 0.6259 vs. 1.4974). [...] **Clustering Models:** **K-Means Clustering** groups users based on behavior and characteristics. User vectors encapsulate attributes for clustering. [...] **Hybrid Approach:** Combines initial clustering with regression for predictive insights and refinement. For example, K-Means segments users, and regression predicts metrics for each segment. [...] **Case Study:** The **Bike Sharing Dataset** demonstrates the effectiveness of hybrid models, with decision trees outperforming linear models. [...] **Conclusion:** Hybrid models provide deep insights into user behavior, enabling personalized services and improved business performance. [...]
AcademicRAG	**Hybrid Models for Enhancing User Segmentation in Ride-Sharing** Hybrid models combining regression and clustering offer comprehensive user segmentation for ride-sharing platforms. Four approaches are highlighted: 1. **Regression + Clustering:** Predict user behaviors (e.g., ride frequency) and cluster based on predictions. Benefits include personalization and operational efficiency. [...] 2. **Clustering + Regression:** Segment users initially with clustering, then refine with regression for dynamic insights. [...] 3. **Hierarchical Clustering + Regression:** Create multi-level segmentation for granular insights and scalability. [...] 4. **Ensemble Methods:** Combine multiple models for robust and accurate predictions. [...] **Practical Example:** A ride-sharing company uses hybrid models to segment users based on ride frequency and preferred times. Steps include data collection, clustering, regression, and implementation. [...] **Conclusion:** Hybrid models enable deeper insights into user behavior, leading to personalized services and improved operational efficiency. [...]
LLM Decision	**Comprehensiveness:** ”Winner”: ”Answer AcademicRAG” Reasoning: AcademicRAG explores multiple hybrid approaches and includes a practical example, while GraphRAG focuses on one approach. [...] **Diversity:** ”Winner”: ”Answer AcademicRAG” Reasoning: AcademicRAG presents diverse methodologies (e.g., hierarchical clustering, ensemble methods), while GraphRAG centers on a single approach. [...] **Empowerment:** ”Winner”: ”Answer AcademicRAG” Reasoning: AcademicRAG provides actionable steps (e.g., model evaluation, implementation), while GraphRAG lacks concrete guidance. [...] **Overall Winner:** ”Winner”: ”Answer AcademicRAG” Reasoning: AcademicRAG excels in comprehensiveness, diversity, and empowerment, making it the superior response. [...]

Table A.3: Case Study: Comparison Between LightRAG and AcademicRAG

Query	What hybrid models (e.g., regression + clustering) enhance user segmentation in ride-sharing?
LightRAG	**Hybrid Models for Enhancing User Segmentation in Ride-Sharing** User segmentation in ride-sharing can be significantly enhanced by combining regression and clustering models. Key approaches include: **1. Regression Models** for Predictive Analytics: Predict bike usage patterns using datasets like bike sharing, incorporating features like weather and seasonal information. [...] **2. Clustering Models** for Segmentation: Apply K-means clustering to group users based on behavior, such as spending patterns or ride preferences. [...] **3. Combining Regression and Clustering:** Use regression to derive features for clustering, then apply regression within clusters for deeper insights. [...] **4. Performance Evaluation:** Use cross-validation and WCSS to evaluate models and ensure optimal performance. [...] **5. Data Transformation:** Normalize and log-transform data to improve model accuracy. [...] **Application in Ride-Sharing**: Segment users, predict behavior, optimize services, and monitor performance. [...]
AcademicRAG	**Hybrid Models for Enhancing User Segmentation in Ride-Sharing** Hybrid models combining regression and clustering offer comprehensive user segmentation for ride-sharing platforms. Four approaches are highlighted: **1. **Regression + Clustering:**** Predict user behaviors (e.g., ride frequency) and cluster based on predictions. Benefits include personalization and operational efficiency. [...] **2. **Clustering + Regression:**** Segment users initially with clustering, then refine with regression for dynamic insights. [...] **3. **Hierarchical Clustering + Regression:**** Create multi-level segmentation for granular insights and scalability. [...] **4. **Ensemble Methods:**** Combine multiple models for robust and accurate predictions. [...] **Practical Example:** A ride-sharing company uses hybrid models to segment users based on ride frequency and preferred times. Steps include data collection, clustering, regression, and implementation. [...] **Conclusion:** Hybrid models enable deeper insights into user behavior, leading to personalized services and improved operational efficiency. [...]
LLM Decision	**Comprehensiveness:** "Winner": "Answer AcademicRAG" Reasoning: AcademicRAG provides a structured, cohesive explanation with clear benefits and a practical example. [...] **Diversity:** "Winner": "Answer AcademicRAG" Reasoning: AcademicRAG offers diverse methodological approaches tailored to ride-sharing. [...] **Empowerment:** "Winner": "Answer AcademicRAG" Reasoning: AcademicRAG provides actionable insights and a clear implementation roadmap. [...] **Overall Winner:** "Answer AcademicRAG" Reasoning: AcademicRAG excels in comprehensiveness, diversity, and empowerment. [...]

Appendix B

Detailed Evaluation Results for Research Literature Assistant

B.1 Results of Literature Review Phase

For the initial literature review phase, we extracted the introduction sections from these papers and constructed a KG to support broad exploration of the field.

Our system demonstrated strong capabilities across several key functions. When asked to identify research trends and gaps shown in Table B.1, the assistant successfully recognized major directional shifts in the field, including the transition from CNN-based to Transformer-based architectures, the integration of advanced generative priors, and increasing focus on real-world applications. More importantly, it identified specific research gaps such as limited exploration of 3D facial structure integration, insufficient handling of extreme degradation cases, and the lack of unified evaluation standards.

The assistant effectively identified relevant papers for the following deeper investigation (Shown in Table B.2), providing not only titles but also concise explanations of each paper's significance to the research topic. This demonstrates the system's ability to contextualize papers within the broader research landscape, a critical function for literature review.

B.2 Results of Paper In-depth Reading Phase

For the in-depth reading phase, we input complete papers identified in the previous literature review phase, constructing a more detailed KG. When queried for methodological details (Shown in Table B.3), the assistant provided comprehensive information about specific approaches such as ControlNeXt, including its architectural components, key innovations, and performance characteristics. This level of detail

Table B.1: Identify the Research Trend and Gap

Query	Identify the research trend and research gap in the field of Blind Face Restoration.
Response	**Current Research Trends in Blind Face Restoration** 1. Shift Towards Transformer-Based Architectures: Recent research demonstrates a clear trend moving from CNN-based approaches to Transformer-based models for blind face restoration. The BFRFormer model exemplifies this shift, specifically designed to address the limitations of CNNs in modeling long-range dependencies. Key aspects of this trend include[...] 2. Integration of Advanced Generative Priors[...] 3. Focus on Real-World Applicability[...] **Key Research Gaps in Blind Face Restoration** 1. Limited Exploration of 3D Facial Structure Integration[...] 2. Insufficient Handling of Extreme Degradation Cases[...] 3. Lack of Unified Evaluation Standards[...] 4. Limited Exploration of Cross-Modal Restoration[...] 5. Computational Efficiency Challenges[...]

Table B.2: Identify Papers

Query	List at least 10 papers that may help me do research about Blind Face Restoration, output their titles and why they are helpful.
Response	Here are the research papers from the Knowledge Base that are highly relevant for Blind Face Restoration research, along with their key contributions: **1. BFRFormer: Transformer-based generator for Real-World Blind Face Restoration** - Helpful because: Introduces a Transformer-based method (BFR-Former) that addresses CNN limitations in modeling long-range dependencies through aggregated attention modules. Features wavelet discriminator and balanced consistency regulation for improved restoration quality. **2. Blind Face Restoration via Integrating Face Shape and Generative Priors**[...] **3. Towards Real-World Blind Face Restoration with Generative Facial Prior** [...] **4. Towards Real-World Blind Face Restoration with Generative Diffusion Prior**[...] **5. Perception Prioritized Training of Diffusion Models**[...] **10. Control3Diff: Learning Controllable 3D Diffusion Models from Single-view Images**[...] **Key reasons these papers are valuable:** - Multiple approaches to leveraging generative priors (GANs, diffusion models) - Novel architectures addressing specific BFR challenges - Introductions of benchmark datasets and evaluation methods - Technical innovations in attention mechanisms and feature fusion - Comprehensive comparisons against state-of-the-art methods

typically requires careful reading of the entire paper, highlighting the assistant's ability to extract and organize complex technical information.

From Table B.4, we found the system also excelled at comparative analysis, effectively contrasting diffusion-based and GAN-based methods for blind face restoration across multiple dimensions including fundamental approaches, architectural components, and performance characteristics. This comprehensive comparison demonstrates the assistant's ability to synthesize information across multiple papers, a task that traditionally requires significant manual effort.

When asked for research suggestions (Shown in Table B.5), the assistant provided actionable guidance for researchers interested in using generative priors for blind face restoration. The suggestions included specific implementation strategies, potential pitfalls to avoid, and references to relevant papers, demonstrating the system's ability to support research planning.

Finally, the assistant successfully classified papers according to their methodological approaches (Shown in Table B.6), organizing the literature into coherent categories such as diffusion model enhancements, model distillation, and domain-specific applications. This classification provides a valuable overview of the research landscape, helping researchers understand the relationships between different approaches.

Table B.3: Introduce Methodological details

Query	Introduce ControlNeXt in detailed.
Response	**Overview of ControlNeXt** ControlNeXt is an advanced and efficient method for controllable image and video generation, designed as a lightweight plug-and-play module with strong robustness and compatibility. It represents a significant advancement in the field of controllable visual generation, offering improved efficiency and robustness compared to previous approaches like ControlNet. Key characteristics of ControlNeXt include: - Parameter Efficiency: Achieves up to 90% reduction in learnable parameters compared to alternatives - Architecture Refinement: Features a streamlined architecture that minimizes latency and reduces trainable parameters - Compatibility: Works seamlessly with various backbone models including Stable Diffusion 1.5, Stable Diffusion XL, and Stable Video Diffusion - Training Stability: Introduces Cross Normalization (CN) to replace "zero-convolution" for faster and more stable training convergence. **Technical Architecture**[...] **1. Lightweight Module Design**[...] **2. Cross Normalization (CN)**[...] **3. Integration with Base Models**[...] **Key Innovations**[...] **Performance and Applications** ControlNeXt demonstrates strong performance across multiple domains: 1. Image Generation[...] 2. Video Generation[...] 3. Comparative Advantages: - Outperforms ControlNet in training convergence and parameter efficiency - Maintains quality while reducing computational requirements - Provides more stable generation capability (as shown in Fig. 9 comparisons) **Implementation Details**[...]

Table B.4: Methods Comparison

Query	Compare Diffusion-based and Gan-based method for Blind Face Restoration.
Response	**1. Fundamental Approaches - Diffusion Models (e.g., BFRffusion)** - Utilize an iterative denoising process to reconstruct high-quality face images from degraded inputs - Leverage multi-scale feature extraction through modules like SDRM, MFEM, TTPM, and PDUM - Employ Stable Diffusion's generative prior to restore realistic facial details - Operate through a Markovian process of gradual noise removal **- GAN-based Methods (e.g., GFP-GAN)** - Use adversarial training between generator and discriminator networks - Employ pretrained face GANs (like StyleGAN2) as generative facial priors - Typically involve one-step generation rather than iterative refinement - Often incorporate spatial feature transform layers for detail preservation **2. Architectural Components**[...] **3. Performance Characteristics**[...] **4. Training and Computational Requirements**[...] **5. Applications and Limitations**[...]

Table B.5: Research Suggestions

Query	If I want to use generative prior in blind face restoration, give me some suggestions.
Response	**1. Leverage Pretrained Face GANs as Generative Priors** - StyleGAN2 Integration: Multiple papers (Zhu et al., 2022; Wang et al., 2021) demonstrate that pretrained StyleGAN2 can serve as a powerful generative prior. It provides rich facial details (e.g., textures, geometry) through its latent space. For example: - SGPN (Zhu et al., 2022) uses StyleGAN2 to generate realistic facial details while integrating 3D face shape priors. The model maps low-quality inputs to StyleGAN2's latent space via a latent encoder (Eq. 6-7 in SGPN). - GFP-GAN (Wang et al., 2021) employs StyleGAN2 with channel-split spatial feature transform (CS-SFT) layers to modulate features, balancing realism and fidelity (Fig. 2 in GFP-GAN). - Implementation Tip: Use a compact version of StyleGAN2 (channel multiplier=1) to reduce computational costs while retaining prior quality (Wang et al., 2021). **2. Combine Generative Priors with Other Facial Priors**[...] **3. Use Spatial Feature Modulation for Fidelity Control**[...] **4. Optimize for Real-World Degradations**[...] **5. Loss Functions for Balanced Outputs**[...] **6. Avoid Common Pitfalls** - **GAN Inversion Limitations**: Pure GAN inversion (e.g., PULSE) often produces low-fidelity results due to limited latent space expressiveness (Zhu et al., 2022). Instead, use hybrid approaches like GFP-GAN or SGPN. - **Reference Priors**: Methods relying on exemplar images (e.g., GFRNet) are impractical; generative priors offer more flexibility (Zhu et al., 2022).

Table B.6: Cluster Topic

Query	Classify the papers in the knowledge base according to their methods.
Response	**1. Diffusion Model Enhancements** Papers focusing on improving diffusion models through novel training techniques or architectural modifications: **- Perception Prioritized Training of Diffusion Models** - Introduces P2 weighting to prioritize noise levels that teach perceptually rich content (SNR-based weighting) - Demonstrates improved FID/KID scores on FFHQ (20.1 → 18.3 FID) and CelebA-HQ - Key insight: Models learn high-level concepts at low SNR, imperceptible details at high SNR - ControlNet[...] - ControlNeXt[...] **2. Model Distillation**[...] **3. Domain-Specific Applications**[...] **4. Foundational Works**[...] **5. Hybrid Approaches**[...] **Missing Methodologies**[...]

TRITA-EECS-EX-2025:247
Stockholm, Sweden 2025

www.kth.se