Multi-Level Querying using A Knowledge Pyramid

Rubing Chen[1], Xulu Zhang[1], Jiaxin Wu[1],
Wenqi Fan[1], **Xiao-Yong Wei [1]**, **Qing Li[1,*]**
[1]The Hong Kong Polytechnic University

Abstract

This paper addresses the need for improved precision in existing Retrieval-Augmented Generation (RAG) methods that primarily focus on enhancing recall. We propose a multi-layer knowledge pyramid approach within the RAG framework to achieve a better balance between precision and recall. The knowledge pyramid consists of three layers: Ontologies, Knowledge Graphs (KGs), and chunk-based raw text. We employ cross-layer augmentation techniques for comprehensive knowledge coverage and dynamic updates of the Ontology schema and instances. To ensure compactness, we utilize cross-layer filtering methods for knowledge condensation in KGs. Our approach, named PolyRAG, follows a waterfall model for retrieval, starting from the top of the pyramid and progressing down until a confident answer is obtained. We introduce two benchmarks for domain-specific knowledge retrieval, one in the academic domain and the other in the financial domain. The effectiveness of the methods has been validated through comprehensive experiments by outperforming 19 SOTA methods. An encouraging observation is that the proposed method has augmented the GPT-4, providing 395% F1 gain by improving its performance from 0.1636 to 0.8109.

1 Introduction

Significant advancements have been made in Large Language Models (LLMs), including proprietary models like ChatGPT and GPT-4, as well as open-source variants like FLAN (Wei et al., 2021) and LLaMA (Touvron et al., 2023a). These models have demonstrated remarkable achievements across a wide range of general knowledge tasks in areas such as language comprehension (Radford et al., 2018; Hadi et al., 2023), logical reasoning (Kojima et al., 2022; Chang et al., 2024), and complex question-answering (Wu et al., 2023b). However, the one-size-fits-all nature of general LLMs

fails to meet the specific demands for professional or personalized knowledge, such as law (Huang et al., 2023b; Cui et al., 2023) and finance domains (Wang et al., 2023; Xie et al., 2023). A straightforward solution is to utilize Supervised Fine-Tuning (SFT) to tailor LLMs to domain-specific tasks (Ouyang et al., 2022). Examples include AlpaCare (Zhang et al., 2023), Mental LLaMA (Yang et al., 2024a) and Zhongjing (Yang et al., 2024b) in medical domain, BloombergGPT (Wu et al., 2023a), Pixiu (Xie et al., 2023), and DISC-FinLLM (Chen et al., 2023) in financial domain, and DISC-LawLLM (Yue et al., 2023) in law domain. Nonetheless, this risks catastrophic forgetting (Luo et al., 2023) of the general knowledge when tuning the original model, and is prone to model hallucination (Huang et al., 2023a).

To address this challenge, the prevalent alternative of Retrieval Augmented Generation (RAG) is introduced as a means to enhance the domain knowledge comprehension of LLMs (Lewis et al., 2020a). Rather than solely relying on the generation capabilities of LLMs to produce answers, RAG takes a different approach by incorporating information retrieval techniques. It retrieves relevant information from existing resources and utilizes this information to enrich the context of the prompt (Gao et al., 2023). This enables in-context learning (Dong et al., 2023) or few-shot learning (Wang et al., 2020b) to be initiated based on the enhanced context. It makes the LLMs' output more stable, accurate, traceable, and interpretable. However, early implementations of RAG have relied on unstructured textual data chunks, which are often obtained by partitioning each document in the corpora database into segments with a predefined chunk size and thus not organized in a specific way (Ma et al., 2023; Lewis et al., 2020b). To address this limitation, Knowledge Graphs (KGs) have been incorporated into RAG (Baek et al., 2023a; Wu et al., 2023d; Abu-Rasheed et al., 2024). The in-
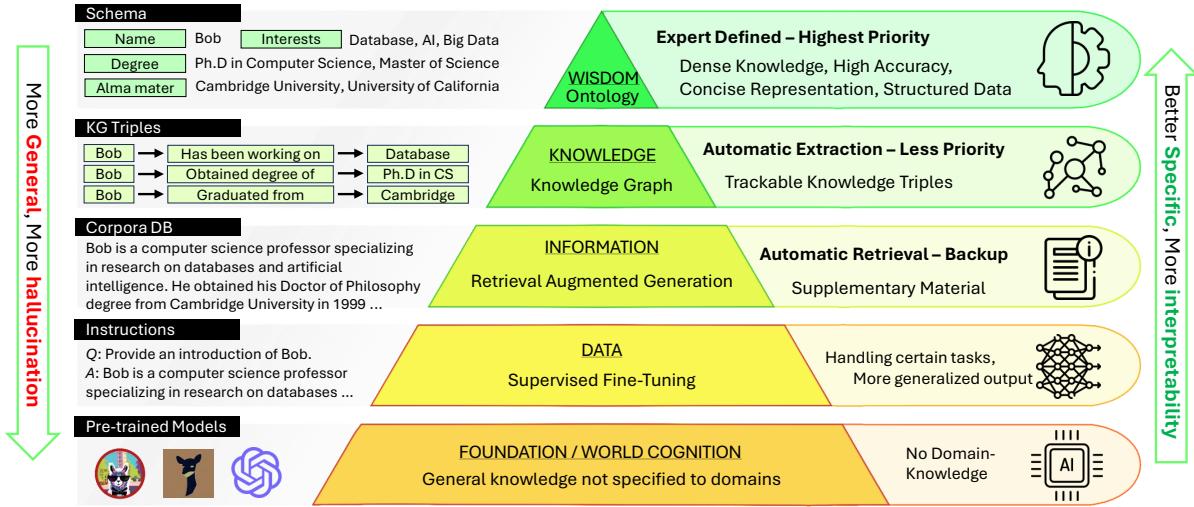
Figure 1: PolyRAG Framework based on the concept of Knowledge Pyramid: different layers signify varying levels of specificity and information richness. The layers are mutually complementary, working together to enhance the overall retrieval and generation process.

corporation of KGs in RAG is a logical choice, as KGs have long been recognized as a promising augmentation method for retrieval tasks. In fact, in traditional retrieval tasks, another solution called Ontologies has been found to be suitable for domain-specific tasks due to their formally and strictly defined schemas (Vamsi et al., 2024; Palagin et al., 2023). However, the use of Ontologies in RAG has been underexplored, partly because the process of defining the Ontology schema requires significant human effort.

In previous RAG methods, the integrated KGs or Ontologies primarily serve the purpose of incorporating additional domain-specific information, thus enhancing the "recall" by increasing the likelihood of locating relevant information. However, the aspect of "precision" has not received specific attention in these methods. For instance, Ontologies are recognized for their precision due to the stringent regulations governing schema and SparQL syntax. Nonetheless, the Ontology search process does not actively participate in the querying process within existing RAG methods. Figure 1 illustrates that different knowledge bases exist at varying levels of specificity, forming a pyramid-like structure. The higher layers of the pyramid contain more specific and structured information, providing a greater likelihood of accurately answering domain-specific questions. Hence, these higher layers are considered more "precision"-friendly. On the other hand, the lower layers of the pyramid are more inclined towards "recall"-friendliness. They possess more

flexible definitions and can store richer information, albeit potentially sacrificing precision.

The observation serves as motivation for us to build a multi-layer knowledge pyramid and devise a customized querying strategy within the RAG framework. The main objective of this approach is to achieve a better balance between "precision" and "recall". Specifically, we organize a knowledge pyramid with three distinct layers of Ontologies, knowledge graphs, and chunk-based raw text. During the construction of the knowledge pyramid, we ensure comprehensive coverage of knowledge by incorporating cross-layer augmentation techniques. This approach reduces the need for human intervention in knowledge completion, as it allows for dynamic updates of the Ontology schema and instances. Moreover, we maintain compactness by employing cross-layer filtering methods, which remove the redundant triplets in the KG for knowledge condensation. The retrieval process follows a waterfall model, starting from the top of the pyramid and progressively moving down until the model reaches a confident answer. If certainty is not achieved, the search continues to the next layer down, and so on. We name it as PolyRAG due to its nature of multi-layer knowledge organization and multi-level querying strategy.

Furthermore, this paper presents a contribution by introducing two benchmarks for domain-specific knowledge retrieval. The first benchmark is specifically designed for the academic domain and has been meticulously constructed by the au-

thors themselves. It encompasses extensive information regarding 1,319 staff members, 2,061 courses, 31 departments, as well as classrooms and facilities at XXX university. In addition, the authors have extended an existing public dataset called FiQA (Shah et al., 2022) in the financial domain to create the second benchmark. We restructured the knowledge from this dataset into a similar pyramid structure, allowing for the application of the proposed methodology. Both benchmarks will be made available to the community.

2 Related Works

2.1 Domain-Specific Large Language Models

LLMs have become integral in various applications, such as chatbots, writing assistants, customer service automation (Sallam, 2023; Rebedea et al., 2023). Domain-specific LLMs are a subset of LLMs that have been tailored to understand and generate text within a particular area of expertise, such as law (Huang et al., 2023b; Cui et al., 2023; Yue et al., 2023), medicine (Zhang et al., 2023; Yang et al., 2024b,a), or finance (Ge et al., 2024; Wang et al., 2023; Xie et al., 2023; Chen et al., 2023). These models aim to provide higher accuracy and more relevant outputs than general-purpose LLMs when dealing with specialized content (Ouyang et al., 2022). A common approach is to adapt instruction fine-tuning based on the general LLMs (Ouyang et al., 2022) using domain-specific copra. However, research has shown that the generality of the fine-tuned model will diminish, resulting in catastrophic forgetting (Luo et al., 2023; Zhai et al., 2023). In order to tackle the limitations, the approach of prompting LLMs with extra domain knowledge as contexts is widely adopted.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances the generative capabilities of language models by incorporating retrieved knowledge for in-context learning (Lewis et al., 2020b; Gao et al., 2023). NaiveRAG represents the most basic architecture within this framework, in which the system retrieves the top-k documents that are most relevant to the query and integrate them into the prompt, thereby grounding the responses in more relevant information (Ma et al., 2023).

Expanding on NaiveRAG, advanced RAG incorporates additional modules or structures to improve retrieval precision. Reranking is a notable example,

where a reranker is employed to refine the initial ranked list (e.g., Re2G (Glass et al., 2022) and bge-reranker (Xiao et al., 2023), both are based on BERT (Kenton and Toutanova, 2019)). Furthermore, studies have indicated that excessive noise and lengthy context can have a negative impact on inference performance. To address this, prompt compression methods such as Selective Context (Litman et al., 2020) and LLMLingua (Anderson et al., 2022) have been developed. These methods emphasize key information while reducing noise and context length, as discussed in (Gao et al., 2023).

2.3 Knowledge-Augmented Language Models

The Knowledge-Augmented Language Model (LM) approach involves integrating LMs with additional knowledge bases to facilitate in-context learning (Liu et al., 2019). In addition to incorporating knowledge in the form of raw texts (Ma et al., 2023), knowledge graphs (KGs) (Baek et al., 2023b; Pan et al., 2024; Wu et al., 2023d) have gained popularity, and Ontologies (Vamsi et al., 2024) have also been utilized, albeit less frequently. In most KG augmentation methods (Pan et al., 2024), the RAG framework is followed, wherein the context is expanded by incorporating retrieved KG triples instead of raw text chunks. KAPING (Baek et al., 2023a) serves as an early example of this approach, which has been later refined in RRA (Wu et al., 2023c). Regarding Ontologies, instead of being used as an individual knowledge base, they are often employed as assistants for generating KG triplets (Pan et al., 2024) (e.g., Text2KGBench (Mihindukulasooriya et al., 2023)) or for augmenting textual corpora (e.g., EKGs (Baldazzi et al., 2023), OntoChatGPT (Palagin et al., 2023)). The integration of various forms of knowledge bases poses a significant challenge and remains an area that has received limited exploration.

3 Knowledge Pyramid Construction

Let us establish the knowledge pyramid as the foundational base for PolyRAG. The construction process begins by creating three distinct knowledge banks, each guided or supported by an Ontology, a knowledge graph, and the raw texts, following common practices. These banks (denoted as $\mathcal{O}$, $\mathcal{K}$, and $\mathcal{T}$, respectively) form the initial layers of the pyramid. The essence of our proposed methodology lies in fostering interactions between these layers,

aimed at enhancing the overall comprehensiveness and compactness of the knowledge base.

3.1 Construction of Initial Layers

Ontology Layer: The Ontology layer $\mathcal{O} = \{\mathcal{O}_s, \mathcal{O}_i\}$ consists of a schema $\mathcal{O}_s$ and corresponding instances $\mathcal{O}_i$. Defining an Ontology schema typically requires significant time and effort from human experts, and achieving a comprehensive schema can be challenging. To simplify the process, one approach is to initially extract a sub-domain schema from general Ontologies like WordNet (Miller, 1995) or ConceptNet (Speer et al., 2017). This extracted schema can serve as a starting point and be refined using the semi-automatic approach presented in Section 3.2.

With the schema $\mathcal{O}_s$ that we extracted, we can guide LLMs to extract instances from the raw text layer $\mathcal{T}$ for each of the concept-attribute pair $(c, a) \in \mathcal{O}_s$, where c is a concept (e.g., *professor*) and a is one of its attributes (e.g., *research_interest*). This can be written as a prompt function

> $f_{ins}(c, a; p)$: Given a paragraph $\{p\}$ from the $\{domain\}$ domain, please identify instances of the Ontology relationship where a class $\{c\}$ has the attribute of $\{a\}$. Note that the attribute may consist of multiple entities.

Executing this function results in instances that fulfill the specified relationship. By repeatedly applying this function to each paragraph, the set $\mathcal{O}_i$ is constructed as

$$\mathcal{O}_i = \{f_{ins}(c, a; p) \,|\, \forall (c, a) \in \mathcal{O}_s, \, \forall p \in \mathcal{T}\}. \quad (1)$$

It is important to note that the specific implementation of the prompt may vary among LLMs. Additionally, in certain cases, it may be beneficial to provide examples to initiate few-shot learning for extracting high-quality information, depending on the capabilities of the LLMs. Our implementations are available in the Github repository.

Knowledge Graph Layer: We adopt Open Information Extraction (OpenIE) (Etzioni et al., 2008) to extract KG triples from raw texts. However, direct extraction often results in noisy output, including irrelevant or duplicate entities. To address this, we draw inspiration from the multi-round prompting approach of LLM2KG (Carta et al., 2023) and reimplement it by introducing four functions: $f_{par}(p)$

for paraphrasing, $f_{ent}()$ for entity extraction, $f_{rel}()$ for relation completion, and $f_{dis}()$ for disambiguation. These functions form a cascade for KG triplet extraction and refinement as

$$f_{kg}(p) = f_{dis}\big(f_{ent}(f_{par}(p)), \, f_{rel}(f_{par}(p))\big). \quad (2)$$

It constructs the initial knowledge graph layer as

$$\mathcal{K} = \{f_{kg}(p) \,|\, \forall p \in \mathcal{T}\}. \quad (3)$$

Due to space limitations, the implementations of these functions are provided in the Appendix A.

3.2 Knowledge Completion

As shown in the pyramid of Figure 1, the higher layers are more structured but not easy to define or extract, resulting in limited coverage. One common issue is the absence of important classes or attributes from the expert-defined Ontology schema, which has historically hindered the ease of defining Ontologies. By utilizing the pyramid framework, we can address this issue in a data-driven manner. The idea is to identify noteworthy concepts and relations that exist in the lower layers but are absent from the higher layers. These identified elements are then incorporated into the higher layers to enhance knowledge completion. For instance, our research reveals that the relationship *publications_in_important_journals* appears frequently in both the knowledge graph and raw texts but is absent from the Ontology. This is due to the tendency of human-defined schema to overlook this specific relationship, as the presence of the *publications* attribute may make it seem redundant. However, users often query this specific relation, and it is explicitly mentioned in professors' profiles. It is thus worth including it in the Ontology.

Our approach involves modeling the semantic distributions of the Ontology and knowledge graph to identify concepts and relations that exhibit significant divergence between the two layers. To achieve this, we begin by transforming class-attribute pairs in the Ontology layer $\mathcal{O}$ into subject+relation phrases that align with the format in the knowledge graph layer $\mathcal{K}$. Instructor embedding (Su et al., 2022) of phrases at both layers are then encoded so as to project them into a common semantic space. We then learn for each layer a multivariate Gaussian using

$$F(\mathbf{X}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{\Sigma}^{\frac{1}{2}}|} e^{-\frac{1}{2}(\mathbf{X}-\mu)^{\top} \mathbf{\Sigma}^{-1}(\mathbf{X}-\mu)}, \quad (4)$$
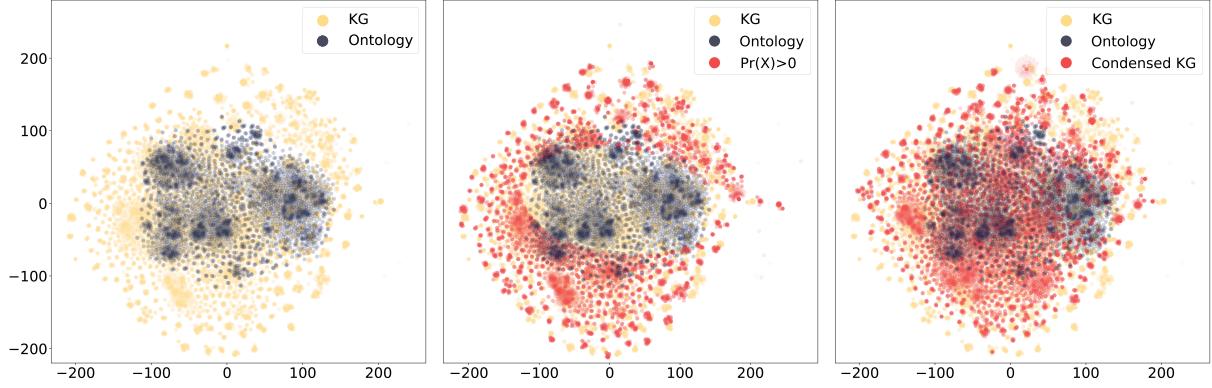
Figure 2: The distribution of KG and Ontology: the original distributions (left), samples with a high priority for knowledge completion are highlighted in red (middle), and the condensed KG after knowledge condensation (right).

where n is the dimension of the space, $\mathbf{X}$ denotes a phrase embedding, and μ and $\mathbf{\Sigma}$ are the mean and covariance matrix, respectively. The semantic distributions of the Ontology and knowledge graph are then modeled as $\mathbf{X}_{\mathcal{O}} \sim \mathcal{N}(\mu_{\mathcal{O}}, \mathbf{\Sigma}_{\mathcal{O}})$ and $\mathbf{X}_{\mathcal{K}} \sim \mathcal{N}(\mu_{\mathcal{K}}, \mathbf{\Sigma}_{\mathcal{K}})$, respectively. The Kullback–Leibler (KL) divergence can then be measured on each given phrase embedding $\mathbf{X}$ as

$$D_{\mathbf{X}}(\mathcal{O} \parallel \mathcal{K}) = -F_{\mathcal{O}}(\mathbf{X}) \log(F_{\mathcal{K}}(\mathbf{X})) \quad (5)$$

$$D_{\mathbf{X}}(\mathcal{K} \parallel \mathcal{O}) = -F_{\mathcal{K}}(\mathbf{X}) \log(F_{\mathcal{O}}(\mathbf{X})), \quad (6)$$

where $D_{\mathbf{X}}(\mathcal{O} \parallel \mathcal{K})$ measures the relative divergence at the point $\mathbf{X}$ by the distribution of $\mathcal{K}$ as the reference, while in $D_{\mathbf{X}}(\mathcal{K} \parallel \mathcal{O})$, the distribution of $\mathcal{K}$ is used as the reference.

At a specific location $\mathbf{X}$, we define a function to rate the priority to extract entities from the knowledge graph to the Ontology as

$$Pr(\mathbf{X}) = D_{\mathbf{X}}(\mathcal{O} \parallel \mathcal{K}) - D_{\mathbf{X}}(\mathcal{K} \parallel \mathcal{O}). \quad (7)$$

The rating outcomes are depicted in Figure 2, illustrating that this function effectively identifies areas where the knowledge graph contains dense information that is comparatively lacking in the Ontology. We then apply k-medoids clustering on knowledge graph triplets within these area to locate new Ontology class and attributes that can be integrated into the Ontology schema, along with corresponding instances. This process can be performed iteratively, allowing for continuous refinement and improvement. As a result of these interactions between the two layers, the comprehensiveness of the pyramid is progressively enhanced.

3.3 Knowledge Condensation

The process of Knowledge Completion aims to enhance the richness of information in a bottom-up manner. In addition, we perform Knowledge Condensation to improve the compactness of the pyramid in a top-down approach. The principle is to leverage the well-structured knowledge in the higher layers to eliminate redundant information present in the lower layers. This is implemented by using each class-attribute in the Ontology layer as an anchor and explore its neighborhood to identify a set of k nearest knowledge graph triplets $\{\mathbf{X_k}\}$. A prompt function is then applied to instruct the LLM to summarize them into compact ones as

> $f_{con}(\{\mathbf{X_k}\})$: Please condense the set of knowledge graph triplets $\{\mathbf{X_k}\}$ obtained from the {*domain*} domain by eliminating redundant triplets and summarizing the remaining ones in a more concise manner. Here are some examples for your reference: {*examples*}. The condensation process should follow the logic of {*CoT*}.

By repeating this process for all anchors, we obtain the condensed knowledge graph layer, as depicted in Figure 2. The original triplets such as "Prof. A works in Z" and "Prof. A is with Z institue" have been merged into more compact form as a relation of "is_a_member_of". It is evident that the compactness of the pyramid has been significantly improved.

4 Multi-Level Querying

The pyramid enables PolyRAG in a straightforward top-down querying manner, following the flow Ontology→Knowledge Graph→Raw Texts, in which the results are returned if answers are found at higher layers; otherwise, the querying continues to the next layer. The querying flow is

Algorithm 1 Multi-Level Querying for PolyRAG

```
 1: function RETRIEVEPOLYRAG(Q)
 2:     qres ← Query_onto(Q, schema)
 3:     C ← Onto(qres)
 4:     if C is empty then
 5:         triples ← Emb_kg(Q, K)
 6:         agreement ← Query_kg(Q, triples)
 7:         if agreement then
 8:             C ← triples
 9:         else
10:             C ← Emb_rag(Q, T)
11:         end if
12:     end if
13:     return C
14: end function
```

depicted in Algorithm 1. Two essential components to build to support the logic are the ways of searching at each layer and conditions to dive next. The Raw Texts layer can be effectively addressed by employing the NaiveRAG approach. Regarding the design of the approaches for the remaining two layers, we propose the following strategies.

4.1 Search at Ontology Layer

At the Ontology Layer, we can utilize SparQL as the query language. SparQL is a well-defined language that provides precise results when queries are properly structured. To achieve this, we can guide the LLM to generate the SparQL query by using the following prompt function

> $Query_{onto}(Q, schema)$: Given a input question $\{Q\}$ within the domain $\{domain\}$, please formulate a SPARQL query to retrieve the answer based on the provided Ontology schema. The namespace is $\{schema.base\}$, and the classes are $\{schema.class\}$. The object properties between classes are $\{schema.op\}$, and the classes may also have data properties such as $\{schema.dp\}$.

For the example of a query question like "Who is currently working in CS Department and was graduated from Cambridge University?", the LLM extracts the related attributes of "works_in" and "graduated_from" with the given $schema$, and generate a SPARQL query to search a staff who satisfies the relevant condition in $\mathcal{O}$. The executing results may include a list of names, or simply

empty due to either the lack of knowledge in Ontology layer, or the mistake of providing unavailable SPARQL due to the limitation of LLM itself.

4.2 Search at Knowledge Graph Layer

At the knowledge graph layer, we employ a retrieval approach similar to the embedding-based retrieval utilized in NaiveRAG. However, instead of using chunks, we work with triplets. Once the matching triplets are retrieved, we utilize a prompt function to assess the Language Model's agreement on whether the question has been answered adequately as

> $Query_{kg}(Q, triples)$: Given a question $\{Q\}$ and the context information provided by the matched triples from the knowledge graph $\{triples\}$, please justify whether the provided information is sufficient to accurately answer the question. Respond with either "Yes" or "No" to provide your justification.

For example, given a query of "which CS staff has interest in cloud computing?", the LLM may justify a set of triples as "agree" if they include information like "Prof. A works in CS Department, Prof. A published on cloud computing journal, ... etc."; inversely, the justification might be "disagree" if the triples do not provide the key knowledge related to this query, thus the searching will process to the next layer for a more relevant context in order to answer the query.

5 Experiments

5.1 Experimental Setup

Our experimentation involves two distinct domain-specific benchmarks. The first benchmark is Academia Challenge (AcadChall), which has been built by ourselves and focuses on the academic domain. It encompasses a comprehensive collection of data obtained from XXX university, including information about 31 departments, 1,319 faculty members, and 2,061 courses. Specifically, AcadChall consists of 512 MCQ and MAQ questions that cover topics related to teaching and research. These questions are intentionally designed to be more challenging compared to existing benchmarks, as each question presents eight answer choices. This design aims to provide a more rigorous assessment of the models' precision. The second benchmark is FiQA, which is widely rec-

Table 1: Performance comparison with SOTA methods on the AcadChall and R-FLUE-FiQA datasets.

Group	Method	Knowledge Base	Backend LLM	AcadChall			R-FLUE-FiQA			
				Precision	Recall	F1	BLEU-2	BLEU-4	BERT	HR(%)
Frozen-LLM	Direct Querying	Pretrained	LLaMA-7B	0.2892	0.5796	0.3470	0.4166	0.2612	0.4658	4.4914
	Direct Querying	Pretrained	LLaMA2-7B	0.2400	0.4039	0.2558	0.4106	0.2533	0.4540	4.8735
	Direct Querying	Pretrained	Vicuna-7B	0.3195	0.6622	0.3969	0.3552	0.2251	0.4509	4.7271
	Direct Querying	Pretrained	Vicuna-13B	0.3044	0.5423	0.3606	0.4098	0.2616	0.4933	5.6721
	Direct Querying	Pretrained	GPT-3.5	0.3716	0.4031	0.3677	0.2597	0.1525	0.5426	4.1965
	Direct Querying	Pretrained	GPT-4	0.1986	0.1453	0.1636	0.4109	0.2457	0.5415	7.6901
SFT	Finance-LLM	Raw Texts	LLaMA-7B	-	-	-	0.3990	0.2428	0.4265	3.8447
	Finance-Chat	Raw Texts	LLaMA2-7B	-	-	-	0.3669	0.2265	0.4531	5.0571
	LoRA	Raw Texts	Vicuna-13B	0.3234	0.6794	0.3914	0.3768	0.2382	0.4511	4.5854
Naive RAG	In-Context	Raw Texts	Vicuna-7B	0.3621	0.8386	0.4673	0.3159	0.2208	0.5858	7.5769
	In-Context	Raw Texts	Vicuna-13B	0.3665	**0.9043**	0.4923	0.2934	0.1866	0.5337	5.4326
	In-Context	Raw Texts	GPT-3.5	0.5105	0.5791	0.5315	0.3618	0.2147	0.5337	3.9153
	In-Context	Raw Texts	GPT-4	0.5032	0.5094	0.4651	0.3982	0.2404	0.5381	5.9918
Advance RAG	Colbertv2	Raw Texts	Vicuna-13B	0.3384	0.6693	0.3815	0.4229	0.2979	0.5918	7.1836
	Bge-reranker	Raw Texts	Vicuna-13B	0.3003	0.6400	0.3586	0.4400	0.3092	0.5965	7.4097
KG Augmentation	KAPING	KG	Vicuna-7B	0.4365	0.8581	0.5133	0.4079	0.2610	0.5389	5.5468
	KAPING	KG	Vicuna-13B	0.3446	0.8945	0.4742	0.3832	0.2403	0.5422	5.4149
	KAPING	KG	GPT-3.5	0.7326	0.6340	0.6433	0.2599	0.1742	0.5300	4.8785
	KAPING	KG	GPT-4	0.6327	0.7065	0.6238	0.3554	0.2319	0.5187	6.1383
Ours	PolyRAG	Pyramid	Vicuna-7B	0.5306	0.8337	0.5967	0.4485	**0.3235**	0.6008	8.6802
	PolyRAG	Pyramid	Vicuna-13B	0.4330	0.8148	0.5655	0.4444	0.3172	0.6089	8.6954
	PolyRAG	Pyramid	GPT-3.5	0.8667	0.7860	0.8039	**0.4459**	0.3204	**0.6114**	**8.7046**
	PolyRAG	Pyramid	GPT-4	**0.8711**	0.7992	**0.8109**	0.4002	0.2609	0.6012	7.7705

Table 2: The impact of Knowledge Completion (CPL) and Condensation (CND) by F1 scores.

Method	Vicuna-7B	Gemini-Pro	GPT-3.5	GPT-4
Naive RAG	0.4673	0.4214	0.5315	0.4651
Baseline	0.5734	0.6899	0.7905	0.7884
+ CPL	0.5874	0.7869	0.7939	0.7763
+ CND	0.5766	0.7439	0.7943	0.7873
+ CPL + CND	**0.5967**	**0.8107**	**0.8039**	**0.8109**

ognized and utilized in the finance domain. Our experimentation follows the dataset split provided by FLUE benchmark (Shah et al., 2022), resulting in the creation of the R-FLUE-FiQA dataset comprising 1,705 open-ended questions. To enhance knowledge completion and condensation, we employ the methods proposed in Section 3 to construct Ontologies, knowledge graphs, and pyramids based on the two datasets. For more detailed information about the two benchmarks, please refer to the Appendix B.

We employ instructor-xl (Su et al., 2022) as our vector base for embedding search and the cosine similarity is used as the metric (Wilkinson and Hingston, 1991). The k values for KG layer and raw texts layer are empirically set to 10 and 5, respectively.

5.2 Evaluation Metrics

Our evaluation process encompasses several metrics to assess the performance, including: 1) For MCQs and MAQs, we employ metrics such as Precision, Recall, and F1 score. 2) For open-ended questions, we utilize BLEU and BERT similarity, calculated based on MiniLM embeddings (Wang et al., 2020a). 3) We introduce a novel metric called HitRate (HR), which quantifies the proportion of correct entities present in the response. These metrics collectively allow for a comprehensive assessment of the system's effectiveness across different question types and response formats.

5.3 Comparison to SOTA Methods

We compare PolyRAG with five groups of state-of-the-art (SOTA) methods, including 1) Frozen-LLMs that are pretrained LLMs of frozen parameters and no external knowledge has been used for querying. 2) SFT (Supervised Fine-Tuning) that are LLMs that undergo supervised fine-tuning on domain-specific datasets. We include two latest implementation of Finance-LLM and Finance-Chat (Cheng et al., 2024), both of which are trained on financial corpora. Additionally, we explore variants of SFT that employ LoRA (Hu et al., 2022) instead of the entire models. 3) NaiveRAG proposed in (Lewis et al., 2020a). 4) Advanced RAG that include two recent models, namely ColBERTv2 (Santhanam et al., 2022) and Bge-reranker (Xiao et al., 2023). 5) KG-Augmented LLMs that consists of models proposed in (Baek et al., 2023b) that integrate knowledge graphs (KGs) into the LLMs. Various LLMs have been explored as the back-

Table 3: The influence of different knowledge layers on the AcadChall benchmark.

Pyramid Layer	Vicuna-7b			Gemini-Pro			GPT-3.5			GPT-4		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Ontology	0.4383	0.6242	0.4445	0.3386	0.3089	0.3157	0.4107	0.3945	0.3979	0.3500	0.4369	0.3677
KG	0.4365	0.8581	0.5133	0.6388	0.4591	0.5099	0.7326	0.6340	0.6433	0.6327	0.7065	0.6238
Raw Texts	0.3621	0.8386	0.4673	0.5152	0.3968	0.4214	0.5105	0.5791	0.5315	0.5032	0.5094	0.4651
Ontology + Raw Texts	0.4535	0.8393	0.5368	0.5905	0.4662	0.5013	0.6890	0.5818	0.6020	0.6464	0.6779	0.6303
KG + Raw Texts	0.3673	0.8758	0.4840	0.6890	0.5428	0.5821	0.8339	0.6622	0.7071	0.8187	0.7729	0.7636
Ontology + KG	0.5251	0.8266	0.5608	0.7755	0.6697	0.6709	0.8567	0.7710	0.7939	0.6083	**0.9032**	0.6871
PolyRAG	**0.5306**	**0.8337**	**0.5967**	**0.8372**	**0.8419**	**0.8107**	**0.8667**	**0.7860**	**0.8039**	**0.8711**	0.7992	**0.8109**

bones, including LLaMA (Touvron et al., 2023a), LLaMA2 (Touvron et al., 2023b), Vicuna (Chiang et al., 2023), Gemini (Team et al., 2023), GPT-3.5 and GPT-4. By combining them with the five groups and PolyRAG, this results in a comprehensive set of 23 methods for performance comparison.

The results are shown in Table 1. It is clear that PolyRAG surpasses other SOTA methods across the five groups, different backbones, and both benchmarks. An encouraging finding is that PolyRAG demonstrates a superior balance between precision and recall. Among the improvements of PolyRAG over others, the gain on the precision ($22.0\% \pm 17.3$) is more pronounced than that of recall ($18.4\% \pm 18.6$). With the incorporation of PolyRAG, GPT-4 (GPT-3.5) has achieved the highest gains by 338.6% (133.2%) over its frozen model and 73.1% (69.8%) over its NaiveRAG run on the AcadChall benchmark. The improvement may result from the fact that around 43.7% (38.4%) of questions are addressed at the Ontology layer, and around 42.3% (44.8%) are at the KG layer on the AcadChall (R-FLUE-FiQA) benchmark.

5.4 Knowledge Completion and Condensation

We have conducted experiments to evaluate the effectiveness of knowledge completion (CPL) and condensation (CND) by integrating them with a baseline model (PolyRAG without these two modules). The results are presented in Table 2. Please note that due to space limitations, we only list results for four specific backbones. For a comprehensive performance analysis with Vicuna-13B and LLaMA2-70B backbones, please refer to the Appendix C. The results clearly demonstrate the effectiveness of both completion and condensation in enhancing performance, whether used individually (CPL: $3.8\% \pm 6.1$ performance gain, CND: $2.2\% \pm 3.3$ performance gain) or in combination ($6.5\% \pm 6.4$ performance gain over the baseline). Our statistics indicate that with completion, queries resolved at the Ontology layer is increased by 5%.

With condensation, more than 27% of queries are resolved at the KG layer.

5.5 Influence of Knowledge Layers

We have examined the influence of different knowledge layers on various backbones. The results are presented in Table 3. It is obvious that the Ontology layer plays a prominent role as a knowledge base in enhancing precision, as evidenced by a precision gain of $25.8\% \pm 7.4$ when combined with the raw text layer and $13.7\% \pm 10.3$ when combined with the KG layer. On the other hand, the KG layer demonstrates a balanced impact on improving either precision or recall. These observations align with our early discussions.

6 Conclusion

This research has described a preliminary investigation that provides a multi-level query approach to domain-specific question answering. In this context, we addressed two significant challenges: the problem of prioritizing knowledge in a certain area and the presence of noise in retrieval situations. In order to overcome the limitations, we introduce Knowledge Pyramid multi-level retrieval framework as PolyRAG. PolyRAG utilizes a sequential retrieval approach and prioritizes knowledge extraction. This approach is designed to tackle the challenges posed by the high demand for dense knowledge in domain-specific scenarios and the distractions caused by noisy contexts. The accuracy of PolyRAG has been validated through extensive tests done in AcadChall and R-FLUE-FiQA, surpassing earlier approaches and generating state-of-the-art results.

Limitations

One limitation to consider is that Language Models (LLMs) do not strictly adhere to the SparQL syntax, which can result in typographical errors in the queries. Further investigation into this aspect is warranted.

Ethics Statement

This research paper adheres to ethical considerations throughout its methodology and findings. The study primarily focuses on the development and evaluation of a retrieval-augmented generation framework, PolyRAG, for domain-specific knowledge retrieval. The research involves the use of existing datasets and benchmarks, as well as the creation of two new benchmarks in the academic and financial domains.

The authors ensure proper citation and acknowledgment of the data sources to maintain transparency and intellectual property rights. In terms of human subjects, this research does not involve any direct human participation, personal data collection, or sensitive information. Therefore, ethical concerns related to informed consent, privacy, and confidentiality are not applicable in this context.

References

Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge graphs as context sources for llm-based explanations of learning recommendations. *Preprint*, arXiv:2403.03008.

Nathan Anderson, Caleb Wilson, and Stephen D Richardson. 2022. Lingua: Addressing scenarios for live interpretation and automatic dubbing. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 202–209.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023a. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023b. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106.

Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili, and Emanuel Sallinger. 2023. Fine-tuning large enterprise language models via ontological reasoning. In *International Joint Conference on Rules and Reasoning*, pages 86–94. Springer.

Salvatore Carta, Alessandro Giuliani, Leonardo Piano, Alessandro Sebastian Podda, Livio Pompianu, and Sandro Gabriele Tiddia. 2023. Iterative zero-shot llm prompting for knowledge graph construction. *arXiv preprint arXiv:2307.01128*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *Preprint*, arXiv:2310.15205.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. In *The Eleventh International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *Preprint*, arXiv:2306.16092.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. 2024. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023b. Lawyer llama technical report. *Preprint*, arXiv:2305.15062.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 1:2.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. 2020. Scatter: selective context attentional scene text recognizer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11962–11972.

Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019. Knowledge-augmented language model and its application to unsupervised named-entity recognition. In *North American Chapter of the Association for Computational Linguistics*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F Enguix, and Kusum Lata. 2023. Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *International Semantic Web Conference*, pages 247–265. Springer.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Oleksandr Palagin, Vladislav Kaverinskiy, Anna Litvin, and Kyrylo Malakhov. 2023. Ontochatgpt information system: Ontology-driven structured prompts for chatgpt meta-learning. *arXiv preprint arXiv:2307.05082*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. *arXiv preprint arXiv:2310.10501*.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. Plaid: an efficient engine for late interaction retrieval.

Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. When flue meets flang: Benchmarks and large pretrained language model for financial domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Krishna Kommineni Vamsi, Vamsi Krishna Kommineni, and Sheeba Samuel. 2024. From human experts to machines: An llm supported approach to ontology and knowledge graph construction.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *NeurIPS Workshop on Instruction Tuning and Instruction Following*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020b. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Ross Wilkinson and Philip Hingston. 1991. Using the cosine measure in a neural network for document retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 202–210.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023a. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Tianyu Wu, Shizhe He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023b. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136.

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, J. Ren, Anhuan Xie, and Wei Song. 2023c. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *ArXiv*, abs/2309.11206.

Yike Wu, Nan Hu, Guilin Qi, Sheng Bi, Jie Ren, Anhuan Xie, and Wei Song. 2023d. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *Preprint*, arXiv:2306.05443.

Kailai Yang, Tianlin Zhang, Ziyan Kuang, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024a. Mentallama: interpretable mental health analysis on social media with large language models. pages 4489–4500.

Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. 2024b. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *Preprint*, arXiv:2309.11325.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. 2023. Alpacare: Instruction-tuned large language models for medical application. *arXiv preprint arXiv:2310.14558*.

Appendix

A Knowledge Graph Layer Construction

This section provides the detailed implantations of the four functions for constructing the knowledge graph layer in Section 3.1.

$f_{par}(p)$: Determine the factual claims from a given paragraph $\{p\}$. Put these facts into short phrases with basic grammar. Remember that every statement should have a distinct meaning, and pronouns should be avoided.

$f_{ent}(f_{par}(p))$: Extract the noun entities in phrases from the given sentences $f_{par}(p)$. The entities should not contain any comma, and each entity should be unique during extraction.

$f_{rel}(f_{par}(p))$: Given the reference context $f_{par}(p)$ and relevant entities, complete the relations between two entities. Notice that a triple should only contain one entity as head, one verb or verb phrase as relation, and one entity as tail. Separate the head, relation, and tail with a comma.

$f_{dis}(f_{ent}(), f_{rel}())$: You are given several triples $f_{rel}()$ with their entities $f_{ent}()$. These triples consist of subject-predicate-object elements, separated with a comma, but may contain ambiguities or inaccuracies. Your task is to refine and disambiguate these triples to ensure that they accurately reflect the entities and relationships described in their source texts without duplication or omissions. If the relationships have the same semantic meaning, rewrite the triples with the same relation. If the triple has already been mentioned with the same meaning as previous triples, delete it.

B AcadChall and R-FLUE-FiQA Benchmarks

Table 4 lists the data statistics of two benchmarks. We use Vicuna-13B (Chiang et al., 2023) to build the knowledge pyramid for two datasets according to the methods illustrated in Section 3.1. Specifically, the AcadChall dataset is created by crawl-

Table 4: Dataset statistics of the two benchmarks.

Dataset	AcadChall	R-FLUE-FiQA
Domain	Academia	Finance
#Raw texts	5,019	17,072
Ontologies		
#Instances	25,481	225,085
#KG Triples	27,920	68,737
#QA pairs	512	1,705
QA type	MCQ&MAQ	Open-ended

ing public information from the official websites of various departments at XXX University. It encompasses comprehensive information about 1,319 staff members, 2,061 courses, 31 departments, classrooms, and facilities, with over 5k corpus entries. Taking the staff member category as an example, it includes not only personal information, contact details, positions held at the university, and department affiliations but also specific academic knowledge, such as staff research interests, alma maters, degrees, research experience, and published papers, which is hard to be captured by general LLMs. In addition, we manually create 512 multi-choice QA pairs. Each question corresponds to eight options, and the answer could be one or more options. For example, for the question "Which staff member in the Art department graduated from Cambridge University?", there could be multiple staff in the provided options that meet the information need. For the R-FLUE-FiQA dataset, we adopt the test set from the FLUE benchmark (Shah et al., 2022), which consists of 1,705 open-ended long-form QA pairs.

C Discussion on Knowledge Completion and Condensation

Table 5 verifies the effectiveness of our proposed knowledge completion and condensation techniques over multiple LLM backends. Specifically, the multi-level querying on the knowledge pyramid without completion and condensation techniques is used as the baseline. In the baseline, around 30% of questions are answered by more structured data (i.e., Ontology or KG). After applying the completion and condensation techniques, the percentage increases to 64%. For example, the relation "*has published in*", which frequently appears in the KG layer, has been used to complete the Ontology layer. It has resulted in those questions related to "publications" being answered by the Ontology

Table 5: The impact of Knowledge Completion (CPL) and Condensation (CND) by precison, recall, and F1 scores.

Method	Vicuna-7b			Vicuna-13b			LLaMA2-70b			Gemini-Pro			GPT-3.5			GPT4		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Naive RAG	0.3621	0.8386	0.4673	0.3665	**0.9043**	0.4923	0.5673	0.3893	0.4193	0.5152	0.3968	0.4214	0.5105	0.5791	0.5315	0.5032	0.5094	0.4651
Baseline	0.5306	0.8337	0.5734	0.4230	0.8848	0.5309	0.7661	0.6479	0.6678	0.7956	0.6513	0.6899	0.7855	0.7806	0.7905	0.7661	**0.8206**	0.7639
+ CPL	**0.6888**	0.8325	0.5874	0.4289	0.8045	0.5595	0.8633	**0.6836**	0.7267	0.8217	0.8597	0.7869	0.8044	**0.8152**	0.7939	0.8063	0.7909	0.7763
+ CND	0.6557	0.7599	0.5766	0.4236	0.8075	0.5557	0.8595	0.6445	0.7166	0.8036	**0.8737**	0.7439	0.7985	0.8027	0.7943	0.8435	0.7853	0.7873
+ CPL + CND	0.5306	**0.8337**	**0.5967**	**0.4330**	0.8148	**0.5655**	**0.8838**	0.6821	**0.7380**	**0.8372**	0.8419	**0.8107**	**0.8667**	0.7860	**0.8039**	**0.8711**	0.7992	**0.8109**

Table 6: The influence of different knowledge layers on the AcadChall benchmark.

Data Source	Vicuna-7b			Vicuna-13b			LLaMA2-70b			Gemini-Pro			GPT-3.5			GPT4		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Ontology	0.4383	0.6242	0.4445	0.3363	0.5811	0.3818	0.5549	0.4501	0.4501	0.3386	0.3089	0.3157	0.4107	0.3945	0.3979	0.3500	0.4369	0.3677
KG	0.4365	0.8581	0.5133	0.3446	0.8945	0.4742	0.6666	**0.6490**	0.6080	0.6388	0.4591	0.5099	0.7326	0.6340	0.6433	0.6327	0.7065	0.6238
Raw Texts	0.3621	0.8386	0.4673	0.3665	0.9043	0.4923	0.5673	0.3893	0.4193	0.5152	0.3968	0.4214	0.5105	0.5791	0.5315	0.5032	0.5094	0.4651
Ontology + Raw Texts	0.4535	0.8393	0.5368	0.4100	0.8908	0.5187	0.6198	0.5244	0.5205	0.5905	0.4662	0.5013	0.6890	0.5818	0.6020	0.6464	0.6779	0.6303
KG + Raw Texts	0.3673	**0.8758**	0.4840	0.3747	**0.9050**	0.4919	0.7421	0.5083	0.5505	0.6890	0.5428	0.5821	0.8339	0.6622	0.7071	0.8187	0.7729	0.7636
Ontology + KG	0.5251	0.8266	0.5608	0.4081	0.8814	0.5166	0.6833	0.5980	0.5922	0.7755	0.6697	0.6709	0.8567	0.7710	0.7939	0.6083	**0.9032**	0.6871
PolyRAG	**0.5306**	0.8337	**0.5967**	**0.4330**	0.8148	**0.5655**	**0.8838**	0.6821	**0.7380**	**0.8372**	0.8419	**0.8107**	**0.8667**	0.7860	**0.8039**	**0.8711**	0.7992	**0.8109**

layer. Similarly, with knowledge condensation, an increase of around 27% of questions are answered by the KG layer. For example, as "*has published research in*", "*has published research articles on*", "*has published a paper in*", "*has work published in*" has been condensed to "*has published research in* " or "*has published research articles in*", the number of tokens that sent to LLM for QA is decreased. Having a compact prompt has eased a smaller LLM backend, such as Vicuna-7B, in understanding the context, and we found that it could lead to a better question-answering result. Overall, with the completion/condensation technique, precision is consistently improved over the baseline across different LLM backends. With the collaboration of completion and condensation techniques, the performance boost is more significant.

D Influence of knowledge layers over backbones

Table 6 lists the impact of different knowledge layers on more LLM backbones than the results presented in Section 5.5. The proposed knowledge pyramid significantly improves the precision of the raw text layer across different backbones. With more powerful backbones, such as Genmini-Pro, GPT-3.5, and GPT-4, the performance gains are larger (i.e., over 0.3). Besides, higher precision could be obtained when combining the Ontology layer with the KG or raw texts layer. A similar observation happens when combing the KG layer with raw text or the Ontology layer across different backbones. These are consistent with the observations in Section 5.5.