Learning to Route: A Rule-Driven Agent Framework for Hybrid-Source Retrieval-Augmented Generation

Haoyue Bai
baihaoyue621@gmail.com
Arizona State University
Tempe, Arizona, USA

Haoyu Wang*
haoyu@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Shengyu Chen
shchen@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Zhengzhang Chen
zchen@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Lu-An Tang
ltang@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Wei Cheng
weicheng@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Yanjie Fu
yanjie.fu@asu.edu
Arizona State University
Tempe, Arizona, USA

Haifeng Chen
haifeng@nec-labs.com
NEC Laboratories America
Princeton, New Jersey, USA

Abstract

Large Language Models (LLMs) have shown remarkable performance on general Question Answering (QA), yet they often struggle in domain-specific scenarios where accurate and up-to-date information is required. Retrieval-Augmented Generation (RAG) addresses this limitation by enriching LLMs with external knowledge, but existing systems primarily rely on unstructured documents, while largely overlooking relational databases, which provide precise, timely, and efficiently queryable factual information, serving as indispensable infrastructure in domains such as finance, healthcare, and scientific research. Motivated by this gap, we conduct a systematic analysis that reveals three central observations: (i) databases and documents offer complementary strengths across queries, (ii) naively combining both sources introduces noise and cost without consistent accuracy gains, and (iii) selecting the most suitable source for each query is crucial to balance effectiveness and efficiency. We further observe that query types show consistent regularities in their alignment with retrieval paths, suggesting that routing decisions can be effectively guided by systematic rules that capture these patterns. Building on these insights, we propose a rule-driven routing framework designed specifically for hybrid-source RAG. A routing agent scores candidate augmentation paths based on explicit rules and selects the most suitable one; a rule-making expert agent refines the rules using QA feedback to produce more comprehensive and reliable decision criteria; and a path-level meta-cache reuses past routing decisions for semantically similar queries to reduce latency and cost. Experiments on three QA datasets demonstrate that our framework consistently outperforms static strategies and learned routing baselines, achieving higher accuracy while maintaining moderate computational cost.

CCS Concepts

• **Information systems** → **Data management systems**; • **Computing methodologies** → **Artificial intelligence**;

Keywords

Routing Mechanisms, Database, RAG

ACM Reference Format:
Haoyue Bai, Haoyu Wang, Shengyu Chen, Zhengzhang Chen, Lu-An Tang, Wei Cheng, Yanjie Fu, and Haifeng Chen. 2026. Learning to Route: A Rule-Driven Agent Framework for Hybrid-Source Retrieval-Augmented Generation. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792676

1 Introduction

Large language models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks, particularly in question answering (QA). Yet, despite their strong capabilities, LLMs still struggle in specialized QA scenarios such as enterprise knowledge access or domain-specific expert consultation, where accurate and up-to-date information is required but may not be captured in the model's parametric memory. To address this limitation, Retrieval-Augmented Generation (RAG) has been proposed [3, 12, 15], which enriches LLMs with external knowledge retrieval, thereby improving factual grounding and adaptability.

Existing RAG systems primarily rely on large unstructured corpora, such as Wikipedia or web documents, as their external knowledge base [14, 23, 37]. Beyond documents, researchers have also explored alternative external sources, such as knowledge graphs for structured entity–relation reasoning [16, 18, 27] and semi-structured data like tables or time series [1, 4, 11]. In contrast, *relational databases* deliver precise, up-to-date, and efficiently queryable factual information [6, 10]. They serve as the backbone of critical

*Corresponding Author, the work was accomplished when the first author working as intern in NEC Labs America supervised by the corresponding author



This work is licensed under a Creative Commons Attribution 4.0 International License.
WWW '26, Dubai, United Arab Emirates
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792676

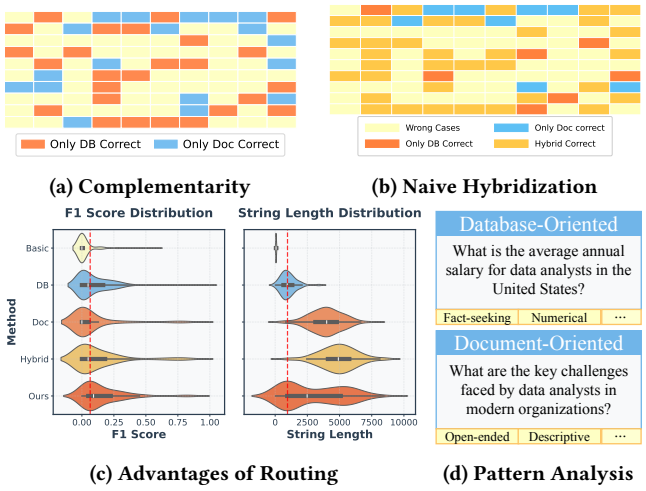


Figure 1: Problem Analysis. (a) Retrieval Path Effectiveness is Question-Dependent. (b) Excess Context Can Hurt Accuracy. (c) Routing Makes Balance Accuracy and Efficiency. (d) Problem types align with database- vs. document-oriented retrieval.

domains such as finance, healthcare, and scientific research, where reliable access to structured records (e.g., financial transactions, electronic medical records, experimental measurements) is indispensable. Unlike unstructured text, databases are designed for accuracy, consistency, and timeliness, ensuring that essential information can be retrieved efficiently and without ambiguity. Despite their broad importance, the integration of relational databases into RAG frameworks has received relatively little attention [28].

To better understand the role of relational databases in RAG, we conduct a set of motivating experiments on the TATQA [39] dataset using GPT-4.1-mini. Here, augmentation follows the standard RAG paradigm: the retrieved evidence is appended to the question and provided to the LLM for answer generation, with the only difference being the external knowledge source. **i) Complementarity.** We first compare two augmentation paths: one enhanced by relational databases (DB), where query results from relational tables are verbalized into text and fed to the LLM, and the other by unstructured documents (Doc), where relevant passages are retrieved from a document corpus. Figure 1a visualizes per-question outcomes, where each cell corresponds to one query and the color denotes which augmentation path yields the correct answer. The results reveal strong complementarity: many questions can only be answered correctly by DB augmentation, while others require Doc augmentation. Neither source dominates across all queries, and relying on a single source leaves substantial coverage gaps. **ii) Naive Hybridization.** A natural follow-up is to combine DB and Doc evidence simultaneously by concatenating both sources and passing them to the LLM. Figure 1b further shows per-question outcomes under this hybrid strategy, where the additional color indicates cases in which both sources are combined. We observe that many queries that can be correctly answered by a single path (DB or Doc) may fail under hybrid augmentation. Instead, redundant or noisy evidence often distracts the model, leading to incorrect answers, while the token count grows dramatically, increasing inference latency and monetary cost. Thus, simply "feeding more" is neither efficient nor reliable. **iii) Necessity of Routing.** Finally, we examine the impact of applying a uniform strategy across all queries. Figure 1c demonstrates that single-source inputs limit accuracy, while hybrid inputs incur high overhead. In contrast, an ideal system should allocate each query to its best-suited path, simultaneously improving accuracy and keeping token usage moderate. This motivates the need for a routing mechanism that achieves a better balance between *effectiveness* and *efficiency*.

Therefore, the key challenge is to design an intelligent routing mechanism that, for each query, can dynamically decide whether to rely on DB, Doc, or both. While prior studies have examined routing in terms of retrieval complexity [13, 19, 32] or retriever choice (e.g., sparse vs. dense) [14, 25, 26, 38], relatively fewer works have addressed routing across heterogeneous knowledge sources, especially bridging structured relational databases with unstructured documents in a unified manner. To ground such a mechanism, it is crucial to understand whether queries exhibit systematic patterns that can guide routing decisions.

Beyond aggregate performance, we further observe that queries are not randomly distributed across sources but instead exhibit clear structural regularities. In particular, fact-centric or numerical questions are typically better served by database augmentation, whereas open-ended or descriptive queries align more naturally with document retrieval, a pattern consistently observed across multiple datasets (see Figure 1d for illustration). However, existing learned routers—whether classifier-based or LLM-based—often struggle to stably capture these heterogeneous patterns. They require large labeled data to train, behave as black boxes during deployment, and tend to produce uncontrollable routing. These limitations motivate a more transparent and rule-grounded approach, where routing decisions are guided by explicit and interpretable rules that encode the observed query–path regularities and capture these understandable patterns. Meanwhile, such a rule-driven perspective also naturally resonates with how humans reason and make decisions. Cognitive studies show that people rarely exhaust all possible options; rather, they often apply simple "if–else" heuristics to make efficient choices under resource constraints [8, 30]. For example, if it rains, people carry an umbrella; if the weather is clear, they leave it at home. These everyday heuristics clearly illustrate how humans rely on interpretable rules to balance accuracy and efficiency without overcomplicating decisions. Motivated by this analogy, we design our framework around *rule-driven routing*, where transparent rules encode observed query-path alignments and are incrementally refined through feedback to balance accuracy, efficiency, and interpretability.

Our framework incorporates three key components: (i) a **Rule-Driven Routing Agent**, which avoids the inefficiency of static augmentation by evaluating candidate paths with interpretable rules and selecting the most suitable one; (ii) a **Rule-Making Expert Agent**, which refines these rules using feedback from QA performance, thereby reducing reliance on static, hand-crafted rules and enabling adaptation to different datasets; and (iii) a **Path-Level Meta-Cache**, which further accelerates inference by reusing routing decisions for repeated or semantically similar queries, eliminating unnecessary agent calls. Together, these components enable adaptive, interpretable, and efficient integration of structured and

unstructured knowledge sources. Our contributions are summarized as follows:

- We conduct analysis of routing between relational databases and document corpora in RAG, highlighting their complementary strengths through motivating experiments.
- We introduce a rule-driven routing framework that integrates a routing agent, a rule-making expert agent, and a path-level meta-cache, aiming to balance accuracy, interpretability, and efficiency.
- We evaluate the proposed framework on three QA benchmarks, showing that it achieves consistent improvements over static strategies and alternative routing baselines, while maintaining moderate cost.

2 Related Work

2.1 Hybrid Knowledge Sources for RAG

Retrieval-Augmented Generation (RAG) is an influential framework that enhances the capabilities of large language models (LLMs) by enabling access to external knowledge sources, thereby improving their performance on tasks that require domain-specific information [3, 12, 15]. Unlike traditional language models, which generate text based solely on internal parameters, RAG models incorporate a retrieval component that fetches relevant content from external data sources and integrates it into the generation process. Traditional RAG systems primarily utilize large-scale unstructured corpora, such as Wikipedia and web documents, as their external knowledge base [14, 23, 37]. Building on this foundation, more recent work has expanded the spectrum of retrieval sources to further enhance RAG's capabilities. Some studies integrate knowledge graphs, which provide structured representations of entities and relationships. Knowledge graphs enable more context-aware and entity-centric retrieval, supporting complex information needs and facilitating multi-hop reasoning [16, 18, 27]. Other approaches incorporate multi-modal data—such as images, videos, and tables—enabling RAG systems to synthesize information across modalities and improving their robustness on heterogeneous, complex queries [1, 4, 11]. In contrast, relational databases store highly structured and precise factual data, supporting efficient updates and fast queries [6, 10]. Despite their widespread use in many enterprise and scientific applications, the integration of relational databases as knowledge sources in RAG has received relatively little attention [28]. There is a lack of systematic analysis and methodical approaches for combining the complementary strengths of relational databases and unstructured documents within RAG frameworks. Addressing this gap, our work aims to explore and propose strategies for the joint utilization of these two data sources.

2.2 Routing Mechanisms in RAG

Routing mechanisms have received increasing attention in RAG systems, enabling dynamic and intelligent decision-making about how each query is processed. Unlike fixed pipelines, routing-enhanced RAG architectures employ dedicated modules that analyze the characteristics of incoming queries and adaptively select the most appropriate retrieval and reasoning pathways, thereby improving both efficiency and response quality. A fundamental role of routing in RAG is to determine the appropriate complexity of the retrieval pipeline for each query. While some questions can be resolved through simple, single-hop retrieval, others require more complex, multi-hop reasoning or iterative retrieval procedures [13, 19, 32]. Adaptive routing allows the system to dynamically decide whether external retrieval is needed or if the language model's parametric knowledge suffices, effectively balancing computational cost and answer accuracy. Another crucial dimension of routing concerns the choice of retrieval methods. Early RAG systems often used simple rule-based or classifier-driven approaches, sending queries with strong lexical overlap to sparse retrievers such as BM25 [25], while directing semantically complex or ambiguous queries to dense retrievers [14]. More recent advances have introduced lightweight classifiers, neural models, and multi-agent frameworks that can select or combine multiple retrieval strategies, sometimes explicitly modeling the interplay between external retrievals and the language model's internal knowledge [26, 38]. Beyond retrieval strategy, routing also governs data source selection. In real-world scenarios, the knowledge necessary to answer a query may reside in document collections, knowledge bases, or other sources. Modern RAG systems increasingly deploy specialized routers—such as domain routers or dynamism routers—to intelligently map queries to the most relevant data sources and to orchestrate hybrid retrieval workflows that capitalize on the complementary advantages of each data type [9, 35, 36]. In this work, we focus on dynamic routing and the integration of structured relational databases with unstructured documents in RAG systems. By recognizing and leveraging the distinct strengths of these sources, we propose a routing mechanism that adaptively selects the knowledge source.

3 Preliminaries

Question-Answering. Given a natural language question q, the most basic question-answering (QA) approach with modern LLMs is to directly generate an answer a based solely on the model's internal (parametric) knowledge, i.e., $a = \mathrm{LLM}(q)$. While extremely efficient, such answers are often limited by the LLM's overall coverage, timeliness, and factual reliability.

Document-Enhanced QA. We then introduce document-enhanced QA, where external unstructured documents are incorporated into the answer generation process. Specifically, for each question q, a retrieval module (e.g., BM25 [25] or dense retriever [14]) searches a pre-built index of document passages, and the top-k retrieved segments $\mathcal{D}_q = \{\mathrm{doc}_1, \ldots, \mathrm{doc}_k\}$ are provided as additional context to the LLM: $a = \mathrm{LLM}(q, \mathcal{D}_q)$. This mechanism allows the LLM to access more domain-specific information during inference.

Database-Enhanced QA. We further consider database-enhanced QA, which leverages structured relational data to provide precise, time-sensitive, and verifiable answers. For a given question q, the system first retrieves relevant tables by matching q against a pre-built index of table metadata, including schemas, high-frequency values, and LLM-generated table descriptions. Once candidate tables are identified, factual records $\mathcal{F}_q$ are extracted from the database either through advanced text-to-SQL generation methods [7, 29] or via efficient keyword-based filtering. The resulting records serve as immutable evidence and are incorporated into the final prompt: $a = \mathrm{LLM}(q, \mathcal{F}_q)$. In our implementation, we explicitly instruct the

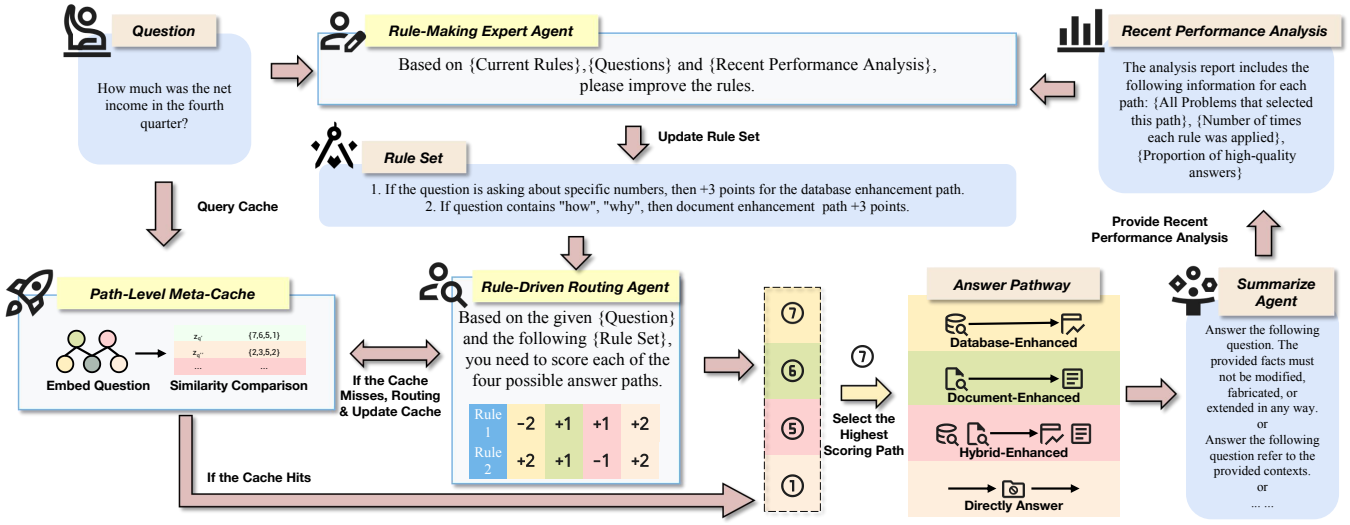


Figure 2: Overall workflow of our rule-driven routing framework. At inference, each question first checks the Path-Level Meta-Cache; if a hit occurs, the cached path scores are reused, otherwise the query is passed to the Rule-Driven Routing Agent, which scores candidate paths based on explicit rules and updates the cache. The highest-scoring path (Database, Document, Hybrid, or Direct Answer) is then selected to provide evidence to the Summarize Agent for answer generation. During training, question–answer performance is collected and analyzed, and the Rule-Making Expert Agent refines the rule set, enabling subsequent routing decisions to adaptively improve.

LLM that retrieved database facts are not to be modified or reinterpreted, ensuring that the generated answers remain precise.

Hybrid-Enhanced QA. We further enable hybrid-enhanced QA. Here, for each query, both the top-ranked document passages $\mathcal{D}_q$ and the extracted database facts $\mathcal{F}_q$ are retrieved and fused as context for the LLM: $a = \text{LLM}(q, \mathcal{D}_q, \mathcal{F}_q)$. This hybridization aims to maximize answer coverage, informativeness, and trustworthiness. However, aggregating more information inevitably increases inference time and computational cost, and in some cases, introducing excessive context may even hurt answer quality.

Routing Mechanisms. The complementary nature of document and database augmentation presents a new challenge: for each question, how to dynamically select the most effective enhancement pathway to balance answer quality, response latency, and computational cost. We formalize this as a query routing problem: given a question q, a routing agent selects an augmentation path $p_q \in \mathcal{P} = \texttt{Doc}, \texttt{DB}, \texttt{Hybrid}, \texttt{LLM}$ according to a policy that optimizes a utility objective combining quality, latency, and cost. The selected pathway determines which information is retrieved and used to construct the prompt for the downstream LLM, which generates the final answer. This design allows the system to flexibly adapt to each query's needs, leveraging the complementary strengths of both data sources while managing efficiency.

4 Methodology

4.1 Overall Framework

Relational databases are well-suited for precise and fast retrieval, especially when queries target clearly structured facts or up-to-date information. However, their coverage is often limited and they struggle with vague, open-ended, or semantic questions. In contrast, document corpora offer broader coverage and are more robust to semantic or fuzzy queries, but they are prone to hallucination and their longer context can increase both latency and cost. To address these trade-offs and enable flexible, high-quality retrieval-augmented QA, we propose a rule-driven routing framework (Figure 2). For each incoming query q, it integrates (i) a Rule-Driven Routing Agent for path scoring and selection, (ii) a Rule-Making Expert Agent for iterative, data-driven rule refinement, (iii) a Path-Level Meta-Cache for efficient and safe decision reuse.

4.2 Rule-Driven Routing Agent

Question intents in such scenarios often follow template-like patterns, which makes them particularly suitable for rule-based routing (e.g., questions starting with "how" typically call for document augmentation). Rule-driven routing is training-free, inherently interpretable, and readily supports domain adaptation and expert intervention. By leveraging language models as the routing agent, the system can operationalize not only simple keyword patterns but also more semantic and context-dependent rules—significantly enhancing flexibility and robustness compared to rigid rule engines. Concretely, the agent takes a query q, evaluates it against a set of rules $\mathcal{R}$, assigns scores to candidate augmentation paths, and selects the highest-scoring path for downstream QA. This design operationalizes human-like heuristics in a principled way, making the routing process flexible, robust, and easy to interpret while retaining the ability to capture nuanced query intents.

Expert-Initialized Rules. Our routing design begins with a set of interpretable rules $\mathcal{R}$, initialized by domain experts. Each rule assigns additive scores to candidate augmentation paths based on recognizable query patterns. Formally, we denote by $S_p(q)$ the score assigned to path p for query q, where higher values indicate stronger preference for selecting that path. A subset of illustrative

rules (complete rules are provided in Appendix A) is as follows:

Rule 1: If query requests numbers, $S_{\texttt{DB}}(q)$+=3.

Rule 2: If query contains "how," "why", $S_{\texttt{Doc}}(q)$+=3.

Rule 3: If query seeks definitions, $S_{\texttt{LLM}}(q)$+=3.

Rule 4: If query seeks fact with explanation, $S_{\texttt{Hybrid}}(q)$+=3.

These rules directly shape routing scores in a transparent manner, ensuring that the reasoning process is both interpretable and auditable. Much like human natural decision-making, such "if–else" heuristics provide simple yet effective guidance.

Routing Mechanism. We introduce a rule-driven routing agent to $\text{A}_{\text{ROUTING}}$ score each candidate path $p \in \mathcal{P}$ for a given query q. The routing policy selects the path with the maximum score:

$$p_q = \arg\max_{p \in \mathcal{P}} S_p(q), \tag{1}$$

where the score is produced by the agent conditioned on both the query and rule set:

$$S_p(q) = \text{A}_{\text{ROUTING}}(q, p, \mathcal{R}). \tag{2}$$

When multiple paths tie for the highest score, a predefined priority order is applied to break ties, ensuring stability and transparency in routing decisions.

Illustrative Example. Consider the query: *"How much was the net income in the fourth quarter?"*. According to Rule 1, the request for a numerical fact linked to a specific time yields an increased score for the DB path. Competing options such as Doc or LLM receive lower scores, as the query is fact-centric rather than descriptive or definitional. Consequently, the agent routes the query to the database path, ensuring both accuracy and efficiency. In practice, the routing agent applies a richer and more rigorous set of rules, which can capture overlapping conditions and subtle query intents.

4.3 Rule-Making Expert Agent

While expert-initialized rules provide a transparent starting point, they may not fully capture the characteristics of a target dataset or benchmark. To reduce reliance on static hand-crafted rules, we introduce a **rule-making expert agent** $\mathcal{A}_{\text{RULE}}$, which refines the rule set using available training or validation data before deployment. The workflow is as follows: after a batch of queries is processed, the system records both routing decisions and QA outcomes. The expert agent then receives the current rule set $\mathcal{R}$ together with a structured summary of these results, and produces an updated rule set $\mathcal{R}^{(t+1)}$ that better aligns routing with utility objectives (e.g., accuracy, efficiency, or their trade-off). This refinement can be carried out offline as a pre-deployment step, ensuring that the rules are better adapted to the specific domain, while keeping the online inference process lightweight and efficient. Much like how humans refine simple heuristics through prior experience before applying them in practice, this step improves the practicality and reliability of rule-driven routing without incurring additional runtime cost.

Rule Performance Diagnostics. After processing a batch of queries, the system generates a diagnostic report to support empirical rule refinement. This report contains four components: (i) the queries themselves, (ii) the current rule set $\mathcal{R}$, (iii) path-level statistics, such as the selection frequency and accuracy of each augmentation path, and (iv) rule-level statistics, including the number of times each rule was triggered and the accuracy achieved when triggered. Together, these diagnostics provide a principled basis for evaluating which rules remain effective, which may require adjustment, and where new rules might be beneficial.

Textual Gradient for Rule Updates. Given the diagnostics $\mathcal{M}^{(t)}$, the expert agent produces refinements to the rule set by interpreting performance feedback in natural language, effectively serving as a textual gradient. This gradient identifies which rules should be strengthened, weakened, or redefined, and proposes modifications accordingly. Formally, rule evolution is defined as:

$$\mathcal{R}^{(t+1)} = \mathcal{A}_{\text{RULE}}(\mathcal{R}^{(t)}, \mathcal{M}^{(t)}), \tag{3}$$

where $\mathcal{A}_{\text{RULE}}$ denotes the expert agent and $\mathcal{M}^{(t)}$ represents the diagnostics at the update step t. By iteratively applying this process, the system refines its routing policy to reflect utility requirements while keeping the online inference process unaffected.

4.4 Path-Level Meta-Cache

While our rule-driven routing agent substantially reduces the inefficiency of naive hybrid augmentation, it still requires an additional agent call for rule evaluation. This extra step introduces some overhead, yet the cost is minor compared to the savings achieved by avoiding redundant hybrid augmentation. Nevertheless, in large-scale deployments, even small latencies can accumulate, motivating further optimization. Since many real-world queries are repeated or semantically similar, caching is a natural choice to further reduce overhead. However, in dynamic scenarios such as frequently updated relational databases, traditional answer-level caches may become unreliable. For example, caching answers to queries like "What is the revenue growth over the past 7 days?" can easily lead to stale or incorrect results as underlying data change. To overcome this limitation, we propose a **path-level meta-cache** that operates at the routing decision level rather than the answer level.

Our meta-cache stores routing decisions—specifically, the embedding representation of each query, the scores for all candidate augmentation paths, and the selected path—for previous queries. This structure enables fast and generalized decision-making for repeated or semantically similar questions while fully preserving the reliability of factual responses. The meta-cache leverages latent-space (embedding) similarity for flexible retrieval, making it particularly well-suited for dynamic or time-sensitive data environments. This design yields three main advantages: (i) reduced computation and latency, by bypassing LLM-based routing for many queries; (ii) semantic generalization, by supporting approximate reuse via embedding similarity; and (iii) robust applicability, as it avoids the pitfalls of stale answer reuse in evolving databases. Formally, let each query q be mapped into a latent representation (embedding) $z_q = \phi(q)$, where $\phi(\cdot)$ denotes the embedding function (e.g., a sentence transformer [24]). The cache is then

$$C = \left\{ (z_{q_j}, S_{\texttt{Doc}}(q_j), S_{\texttt{DB}}(q_j), S_{\texttt{Hybrid}}(q_j), S_{\texttt{LLM}}(q_j)) \right\}, \tag{4}$$

where $S_p(q_j)$ is the score for path $p \in \mathcal{P} = \{\texttt{Doc}, \texttt{DB}, \texttt{Hybrid}, \texttt{LLM}\}$ for previous query q_j. For a new query q', with embedding $z_{q'}$, the system first checks if there exists a cached z_{q_j} such that

$$\text{sim}(z_{q_j}, z_{q'}) \geq \tau, \tag{5}$$

Table 1: Performance Comparison across Models and Methods (F1 / Accuracy)

Method		TATQA		FinQA		WikiQA	
		F1	Acc	F1	Acc	F1	Acc
Basic QA		0.0350	0.050	0.0008	0.010	0.0432	0.138
Doc		0.0608	0.100	0.0022	0.028	0.0679	0.236
DB		0.0530	0.150	0.0032	0.040	0.0513	0.188
Hybrid		0.0872	0.190	0.0016	0.030	0.0899	0.280
Rule-Based		0.0656	0.170	0.0032	0.040	0.0868	0.186
Adaptive-RAG		0.0345	0.180	0.0012	0.008	0.0940	0.158
LLaMA-3	Agent-Based	0.0576 ± 0.0017	0.130 ± 0.0051	0.0017 ± 0.0002	0.014 ± 0.0004	0.0911 ± 0.0028	0.152 ± 0.0049
	Rule Agent	0.0747 ± 0.0112	0.176 ± 0.0050	0.0036 ± 0.0001	0.040 ± 0.0019	0.0812 ± 0.0032	0.234 ± 0.0087
	Score Agent	0.0737 ± 0.0091	0.188 ± 0.0035	0.0037 ± 0.0013	0.038 ± 0.0156	0.0935 ± 0.0031	0.260 ± 0.0093
	Ours	**0.0936** ± 0.0082	**0.212** ± 0.0182	0.0033 ± 0.0002	0.040 ± 0.0056	**0.0987** ± 0.0039	**0.288** ± 0.0074
	Ours-c	0.0900 ± 0.0031	0.194 ± 0.0025	**0.0042** ± 0.0005	**0.046** ± 0.0053	0.0961 ± 0.0021	0.282 ± 0.0151
Qwen2.5	Agent-Based	0.0800 ± 0.0026	0.188 ± 0.0057	0.0037 ± 0.0001	0.046 ± 0.0021	0.0884 ± 0.0033	0.258 ± 0.0069
	Rule Agent	0.0790 ± 0.0023	0.182 ± 0.0051	0.0037 ± 0.0002	0.042 ± 0.0018	0.0815 ± 0.0025	0.240 ± 0.0074
	Score Agent	0.0751 ± 0.0124	0.180 ± 0.0149	0.0038 ± 0.0003	0.042 ± 0.0020	0.0855 ± 0.0026	0.240 ± 0.0063
	Ours	**0.0978** ± 0.0117	**0.220** ± 0.0185	0.0045 ± 0.0005	0.050 ± 0.0043	**0.0970** ± 0.0042	**0.302** ± 0.0092
	Ours-c	0.0944 ± 0.0012	0.218 ± 0.0011	**0.0047** ± 0.0002	**0.054** ± 0.0021	0.0882 ± 0.0024	0.290 ± 0.0064
GPT-4o	Agent-Based	0.0693 ± 0.0007	0.1380 ± 0.0009	0.0040 ± 0.0003	0.0360 ± 0.0028	0.0911 ± 0.0016	0.2020 ± 0.0041
	Rule Agent	0.0800 ± 0.0026	0.1820 ± 0.0074	0.0039 ± 0.0001	0.0420 ± 0.0019	0.0868 ± 0.0014	0.2280 ± 0.0060
	Score Agent	0.0796 ± 0.0014	0.1960 ± 0.0068	0.0037 ± 0.0001	0.0430 ± 0.0009	0.0919 ± 0.0010	0.2480 ± 0.0109
	Ours	**0.0991** ± 0.0048	**0.2200** ± 0.0123	**0.0045** ± 0.0003	**0.0480** ± 0.0021	**0.0930** ± 0.0019	**0.2620** ± 0.0116
	Ours-c	0.0952 ± 0.0041	0.2120 ± 0.0098	0.0042 ± 0.0004	0.0460 ± 0.0017	0.0915 ± 0.0022	0.2500 ± 0.0103
GPT-4.1	Agent-Based	0.0602 ± 0.0028	0.158 ± 0.0087	0.0034 ± 0.0004	0.042 ± 0.0020	0.0905 ± 0.0025	0.214 ± 0.0056
	Rule Agent	0.0709 ± 0.0024	0.180 ± 0.0053	0.0033 ± 0.0005	0.040 ± 0.0019	0.0913 ± 0.0031	0.228 ± 0.0062
	Score Agent	0.0726 ± 0.0029	0.178 ± 0.0044	0.0038 ± 0.0008	0.042 ± 0.0020	0.0915 ± 0.0027	0.208 ± 0.0054
	Ours	**0.0779** ± 0.0037	**0.184** ± 0.0151	**0.0042** ± 0.0012	**0.046** ± 0.0038	**0.0954** ± 0.0054	0.242 ± 0.0082
	Ours-c	0.0754 ± 0.0051	0.178 ± 0.0294	0.0040 ± 0.0007	0.042 ± 0.0024	0.0915 ± 0.0036	**0.246** ± 0.0071
Oracle		0.1020	0.264	0.0046	0.062	0.1050	0.398

where τ is a threshold used to balance the probability and the accuracy of hitting. If so, the routing decision and path scores from the matched entry are directly reused; otherwise, the system falls back to the full rule-driven routing process and updates the cache by adding the current query.

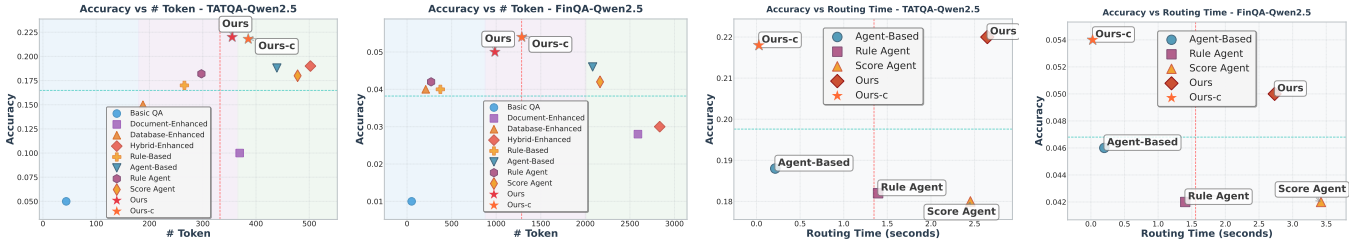
5 Experiment

5.1 Experiment Setting

Datasets. We selected three datasets that include both structured database support and unstructured document support for experiments. TATQA [39] and FINQA [5] are financial report datasets. WIKIQA is a general knowledge question answering dataset. For WIKIQA, We use data from WTQ [21] as the structured dataset and content from Wikipedia [14] as the unstructured text. The test questions are constructed from SQuAD [22] and WTQ [21].

Baselines. We compare our method against two categories of baselines. First, we consider *non-dynamic pathways* that adopt a fixed augmentation strategy: directly answering without retrieval (Basic), document-enhanced retrieval (Doc), database-enhanced retrieval (DB), and hybrid-enhanced retrieval (Hybrid). Second, we evaluate *dynamic routing strategies*, which adaptively select augmentation paths for each query. Among them, lightweight approaches include a keyword-driven rule-based router that maps queries to paths using handcrafted rules [31]. We also include *Adaptive-RAG* [13], which trains a classifier to predict the most suitable path for a given query. Finally, we compare against several LLM-enabled routing methods: (i) *Agent-Based*, where the LLM directly chooses a path from the query [28]; (ii) *Rule Agent*, where the LLM selects paths based on explicit rules [17]; and (iii) *Score Agent*, where the LLM scores rules and selects the path with the highest score [20].

Implementation. We adopt a consistent retrieval setup across all pathways to ensure comparability. Specifically, we use BM25 [25], a classical term-based sparse retrieval model, to retrieve candidate documents or table metadata depending on the selected pathway. For answer generation, we employ GPT-4.1-mini to produce



(a) Accuracy vs #Token - TATQA (b) Accuracy vs #Token - FinQA (c) Accuracy vs Time - TATQA (d) Accuracy vs Time - FinQA

Figure 3: Accuracy and Efficiency Analysis

responses under each augmentation pathway. To guarantee fairness, the routing policy is restricted to selecting among these precomputed answers rather than re-generating them, so that differences stem solely from routing choices rather than generation variability. Following prior work [13, 34], we randomly sample 500 queries for evaluation. In settings that require training data (e.g., for rule refinement), we allocate an additional set of 100 queries as the training set. For both the routing agent and the rule-making expert agent, we experiment with three backbone LLMs: LLaMA-3-8B-Instruct (abbreviated as LLaMA-3) [2], Qwen2.5-14B-Instruct (Qwen2.5) [33], GPT-4o-mini (GPT-4o) and GPT-4.1-mini (GPT-4.1). All experiments are repeated multiple times with different random seeds, and we report the mean and standard deviation across runs.

5.2 Performance Comparison with Existing Methods

This experiment aims to evaluate whether dynamic routing can provide more accurate QA performance compared to single-path and static hybrid strategies. The results in Table 1 highlight two key observations. First, routing-based strategies generally outperform single augmentation paths, demonstrating the advantage of adaptively selecting the most suitable information source. Interestingly, in several cases the routing baselines even surpass the hybrid strategy, indicating that blindly concatenating multiple sources can introduce noise, while effective routing helps identify the correct path and avoid unnecessary context. Second, our proposed framework consistently outperforms all routing baselines across datasets and model backbones. For instance, on TATQA with LLaMA-3, our approach improves accuracy from 0.188 (Score Agent) to **0.212**, and on WikiQA with Qwen2.5 it achieves a new state-of-the-art accuracy of **0.302**, well above the best competing baseline (0.260). These results confirm that our method delivers more accurate and stable performance, validating the effectiveness of the designed routing mechanism.

5.3 Accuracy vs. Token Efficiency

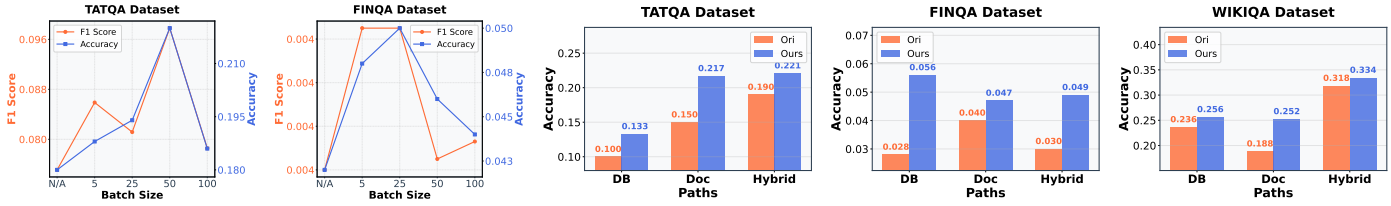
This experiment aims to examine the trade-off between model accuracy and computational cost. Since the number of tokens directly correlates with both inference latency and monetary expense, we adopt it as a proxy measure of efficiency. Figure 3a and 3b show the accuracy-token relationship on TATQA and FinQA under the Qwen2.5 backbone. On TATQA, our methods (Ours and Ours-c) achieve accuracies above 0.21 while keeping token counts around 300, whereas the hybrid strategy consumes over 400 tokens for noticeably lower accuracy. Similarly, on FinQA, both Ours and Ours-c dominate the upper-left region of the plot, offering clear accuracy advantages without the excessive token usage exhibited by document-heavy strategies. We observe that our proposed method achieves the highest accuracy across both datasets, while maintaining only moderate token usage. In contrast, static hybrid strategies often incur large token overhead without clear accuracy gains, and some routing baselines sacrifice performance to reduce cost. These results demonstrate that our framework not only delivers superior QA accuracy, but also sustains a favorable balance between effectiveness and efficiency.

5.4 Accuracy vs. Routing Time

This experiment investigates the fundamental trade-off between routing efficiency and overall QA performance. Routing time is a particularly critical factor in real-world systems, as it directly reflects the latency introduced before answer generation. Figure 3c and Figure 3d present the relationship between accuracy and routing time on TATQA and FinQA using the Qwen2.5 backbone. On TATQA, our method Ours-c attains the highest accuracy while requiring almost negligible routing time, clearly outperforming conventional rule-based or score-based agents that consume over 2 seconds on average with noticeably lower accuracy. Similarly, on FinQA, Ours-c again dominates the upper-left corner, delivering consistently superior accuracy with minimal routing latency, while alternatives such as Score Agent and Rule Agent remain both slower and less accurate. These results confirm that our framework achieves a highly favorable balance by providing higher accuracy at substantially lower routing cost, highlighting its distinctive advantage for deployment in latency-sensitive applications.

5.5 Effect of Rule Update Frequency

This experiment investigates the impact of frequency on the effectiveness of rule updates performed by the expert agent. Since rule updates refine the routing policy based on accumulated query–answer feedback, the frequency of updates determines how quickly the system adapts to data distribution. Figure 4a and Figure 4b present the results on TATQA and FinQA. We first observe that even a single update, corresponding to batch size 100, already improves performance compared with the no-update case. Moreover, moderate batch sizes, such as 25 or 50, further enhance both accuracy and F1 score, showing that more frequent updates help the routing



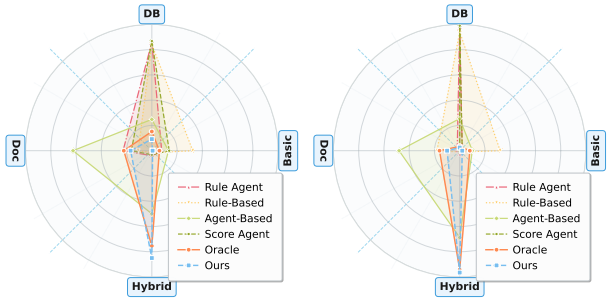
(a) Effect of Rule Update on TATQA (b) Effect of Rule Update on FINQA (c) Path Detail on TATQA (d) Path Detail on FINQA (e) Path Detail on WIKIQA

Figure 4: Investigation of Proposed Method

agent align with empirical outcomes. For example, on TATQA, the F1 score increases steadily from 0.080 without updates to over 0.096 when batch size is set to 50. On FinQA, accuracy peaks when the batch size is 25, indicating that timely adaptation yields the most effective routing policy. These results confirm that incorporating rule updates is consistently beneficial, and that an appropriate update frequency enables the framework to achieve further improvements by better capturing evolving query characteristics.

5.6 Analysis of Path Utilization

This experiment examines how routing affects the effectiveness of individual augmentation paths. In Figure 4c–4e, the bar plots compare two settings: the accuracy obtained when all queries are forced through a single augmentation path (orange), versus the accuracy of the same path when it is selectively chosen by our routing mechanism (blue). This comparison isolates the effect of routing on path quality, independent of the overall distribution of queries across paths. We observe that across all datasets, selectively chosen paths consistently achieve higher accuracy than their single-path baselines. For instance, on TATQA as well as FinQA and WikiQA, the DB, Doc, and Hybrid paths all show clear accuracy gains when routing assigns only the most suitable queries to them. These results demonstrate that routing not only improves overall system accuracy but also enhances each individual augmentation strategy by avoiding inefficient one-size-fits-all usage and instead leveraging path complementarity in a targeted and adaptive manner.



(a) Path Selection on TATQA (b) Path Selection on FINQA

Figure 5: Investigation of Path Selection. For better visualization, the Basic and Doc results are proportionally scaled up while preserving their relative ratios.

5.7 Investigation of Path Selection

To better understand how our method allocates queries across different augmentation strategies, we further analyze the distribution of selected paths. Figure 5 reports the path selection profiles on TATQA and FinQA under the Qwen2.5 backbone. Each radar plot shows the proportion of queries routed to the DB, Doc, Hybrid, and Basic paths, comparing our method with several baselines and an oracle strategy. The oracle represents an upper bound where each query is assigned to the augmentation path that yields the correct answer. We observe that baseline strategies exhibit skewed or inconsistent allocations: rule-based and rule-agent methods often over-assign queries to DB, while agent-based and score-agent approaches tend to misallocate queries to suboptimal paths. In contrast, our method produces a distribution that closely matches the oracle, with a suitable use of DB, Doc, and Hybrid paths according to the query. This demonstrates that our routing mechanism is able to approximate oracle-level path decisions, effectively leveraging the complementary strengths of heterogeneous knowledge sources.

6 Conclusion

In this work, we investigated the complementary roles of relational databases and unstructured documents in retrieval-augmented question answering. Our analyses showed that these sources excel on different query types: databases are most effective for fact-centric and numerical questions, while documents better handle open-ended or descriptive queries. We further demonstrated that naïve hybrid augmentation is not a solution, as it increases token overhead and latency without consistent accuracy gains, underscoring the necessity of routing. To this end, we introduced a rule-driven routing framework that grounds path selection in explicit, interpretable rules, continuously refines them through feedback with a rule-making expert agent, and accelerates inference with a path-level meta-cache. Extensive experiments on three QA benchmarks confirmed that our method consistently outperforms static and learned routing baselines, achieving higher accuracy while controlling cost. Overall, this study highlights the importance of principled routing across structured and unstructured knowledge sources, paving the way for more accurate, efficient, and interpretable RAG systems.

Acknowledgments

This research was done during the internship at NEC Labs America. Dr. Yanjie Fu is supported by the National Science Foundation (NSF) under grant numbers 2426340, 2416727, 2421865, and 2421803.

References

[1] Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, MohammadAli Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. 2025. Ask in Any Modality: A Comprehensive Survey on Multimodal Retrieval-Augmented Generation. *CoRR* abs/2502.08826 (2025). arXiv:2502.08826 doi:10.48550/ARXIV.2502.08826

[2] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 2206–2240. https://proceedings.mlr.press/v162/borgeaud22a.html

[4] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. 2022. MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 5558–5570. doi:10.18653/V1/2022.EMNLP-MAIN.375

[5] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Kenneth Huang, Bryan R. Routledge, and William Yang Wang. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3697–3711. doi:10.18653/V1/2021.EMNLP-MAIN.300

[6] Edgar F Codd. 2007. Relational database: A practical foundation for productivity. In *ACM Turing award lectures*. 1981.

[7] Naihao Deng, Yulong Chen, and Yue Zhang. 2022. Recent advances in text-to-SQL: A survey of what we have and what we expect. *arXiv preprint arXiv:2208.10099* (2022).

[8] Gerd Gigerenzer, Peter M Todd, the ABC Research Group, et al. 2000. *Simple heuristics that make us smart*. Oxford University Press.

[9] Rachid Guerraoui, Anne-Marie Kermarrec, Diana Petrescu, Rafael Pires, Mathis Randl, and Martijn de Vos. 2025. Efficient Federated Search for Retrieval-Augmented Generation. In *Proceedings of the 5th Workshop on Machine Learning and Systems, EuroMLSys 2025, World Trade Center, Rotterdam, The Netherlands, 30 March 2025- 3 April 2025*, Eiko Yoneki and Amir H. Payberah (Eds.). ACM, 74–81. doi:10.1145/3721146.3721942

[10] Jan L Harrington. 2016. *Relational database design and implementation*. Morgan Kaufmann.

[11] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. 2023. Reveal: Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 23369–23379. doi:10.1109/CVPR52729.2023.02238

[12] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *J. Mach. Learn. Res.* 24 (2023), 251:1–251:43. https://jmlr.org/papers/v24/23-0037.html

[13] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (Eds.). Association for Computational Linguistics, 7036–7050. doi:10.18653/V1/2024.NAACL-LONG.389

[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 6769–6781. doi:10.18653/V1/2020.EMNLP-MAIN.550

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[16] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=cPgh4gWZlz

[17] Xinyu Lu, Bowen Yu, Yaojie Lu, Hongyu Lin, Haiyang Yu, Le Sun, Xianpei Han, and Yongbin Li. 2024. SoFA: Shielded On-the-fly Alignment via Priority Rule Following. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 7108–7136. doi:10.18653/v1/2024.findings-acl.424

[18] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=oFBu7qaZpS

[19] Seiji Maekawa, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. Retrieval Helps or Hurts? A Deeper Dive into the Efficacy of Retrieval Augmentation to Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (Eds.). Association for Computational Linguistics, 5506–5521. doi:10.18653/V1/2024.NAACL-LONG.308

[20] Zhihong Pan, Kai Zhang, Yuze Zhao, and Yupeng Han. 2025. Route to Reason: Adaptive Routing for LLM and Reasoning Strategy Selection. *CoRR* abs/2505.19435 (2025). arXiv:2505.19435 doi:10.48550/ARXIV.2505.19435

[21] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 1470–1480. doi:10.3115/V1/P15-1142

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, Jian Su, Xavier Carreras, and Kevin Duh (Eds.). The Association for Computational Linguistics, 2383–2392. doi:10.18653/V1/D16-1264

[23] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Trans. Assoc. Comput. Linguistics* 11 (2023), 1316–1331. doi:10.1162/TACL_A_00605

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. doi:10.18653/V1/D19-1410

[25] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[26] Tolga Şakar and Hakan Emekci. 2025. Maximizing RAG efficiency: A comparative analysis of RAG methods. *Natural Language Processing* 31, 1 (2025), 1–25.

[27] Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF 2024, Brooklyn, NY, USA, November 14-17, 2024*. ACM, 608–616. doi:10.1145/3677052.3698671

[28] Antony Seabra, Claudio Cavalcante, João Nepomuceno, Lucas Lago, Nicolaas Ruberg, and Sérgio Lifschitz. 2024. Dynamic Multi-Agent Orchestration and Retrieval for Multi-Source Question-Answer Systems using Large Language Models. *CoRR* abs/2412.17964 (2024). arXiv:2412.17964 doi:10.48550/ARXIV.2412.17964

[29] Liang Shi, Zhengju Tang, Nan Zhang, Xiaotong Zhang, and Zhi Yang. 2025. A survey on employing large language models for text-to-sql tasks. *Comput. Surveys* 58, 2 (2025), 1–37.

[30] Herbert A Simon. 1990. Bounded rationality. In *Utility and probability*. Springer, 15–18.
[31] Håkan Sundblad. 2007. *Question classification in question answering systems*. Linkopings Universitet (Sweden).
[32] Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small Models, Big Insights: Leveraging Slim Proxy Models To Decide When and What to Retrieve for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 4420–4436. doi:10.18653/V1/2024.ACL-LONG.242
[33] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/
[34] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 10014–10037. doi:10.18653/V1/2023.ACL-LONG.557
[35] Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. 2025. Talk to Right Specialists: Routing and Planning in Multi-agent System for Question Answering. *CoRR* abs/2501.07813 (2025). arXiv:2501.07813 doi:10.48550/ARXIV.2501.07813
[36] Yikuan Xia, Jiazun Chen, Yirui Zhan, Suifeng Zhao, Weipeng Jiang, Chaorui Zhang, Wei Han, Bo Bai, and Jun Gao. 2025. ER-RAG: Enhance RAG with ER-Based Unified Modeling of Heterogeneous Data Sources. *CoRR* abs/2504.06271 (2025). arXiv:2504.06271 doi:10.48550/ARXIV.2504.06271
[37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=WE_vluYUL-X
[38] Jiarui Zhang, Xiangyu Liu, Yong Hu, Chaoyue Niu, Fan Wu, and Guihai Chen. 2025. Query Routing for Retrieval-Augmented Language Models. *CoRR* abs/2505.23052 (2025). arXiv:2505.23052 doi:10.48550/ARXIV.2505.23052
[39] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3277–3287. doi:10.18653/V1/2021.ACL-LONG.254

A Rule Updates

Figures 6–9 present the evolution of routing rules used in our framework. Figure 6 shows the initial hand-crafted rules, while Figures 7, 8, and 9 illustrate the updated rules learned on TATQA, FinQA, and WikiQA, respectively. These results are obtained using the Qwen2.5 backbone and demonstrate how the rule-making expert agent incrementally refines the rule set based on query–answer feedback. The updates capture dataset-specific patterns (e.g., numerical queries in TATQA, financial reasoning in FinQA, and open-domain descriptions in WikiQA), confirming that our rule-driven routing framework can adaptively specialize rules to different domains.

B Prompts

Figures 10 and 11 provide the full prompt templates used in our routing framework. Figure 10 shows the prompt for the score-based routing agent, where the LLM is instructed to evaluate candidate augmentation paths and return numerical scores. Figure 11 shows the prompt for the rule-driven routing agent, where the LLM is guided to interpret rules and apply them to the current query. Together, these prompts define the interaction between the LLM and our routing components, ensuring reproducibility and clarity of the experimental setup.

```
1. If question requests numbers, percentages, years, or calculations, then FACT_ONLY path +3 points
2. If question contains "how", "why", "where", or asks for process/explanation, then DOC_ONLY path +3 points
3. If question asks for definition or general knowledge, then LLM_ONLY path +3 points
4. If question has multiple requirements (both data and explanation), then DUAL path +3 points
```

Figure 6: Initial Rule

```
1. If question requests numbers, percentages, years, or calculations without requiring context or
explanation, then FACT_ONLY path +5 points (strengthened from +3)
2. If question contains "how", "why", "where" or asks for process/explanation, and if the answer requires
detailed context or reasoning, then DOC_ONLY path +5 points (strengthened from +3)
3. If question asks for definitions or general knowledge, especially those that require broad understanding
or interpretation, then LLM_ONLY path +2 points (reduced from +3 due to poor performance, but still given
some weight as certain definitions might benefit from LLMs)
4. If question has multiple requirements (both data and explanation), or requires interpretation alongside
data, then DUAL path +7 points (strengthened from +3)
5. If question is about specific historical events or dates that require referencing documents for context,
then DOC_ONLY path +4 points
6. If question involves complex calculations or data aggregation that may benefit from structured data
sources and some interpretation, then DUAL path +6 points
7. If question is about financial metrics or ratios that require both precise data and some level of business
context, then DUAL path +6 points
```

Figure 7: TATQA Updated Rule

```
1. If question requests numbers, percentages, years, or calculations, then FACT_ONLY path +5 points
2. If question contains "how", "why", "where", or asks for process/explanation, then DOC_ONLY path +5 points
3. If question asks for definition or general knowledge, then LLM_ONLY path +5 points
4. If question involves complex financial data or historical data comparisons (e.g., "change from 2014 to
2015"), then FACT_ONLY path +5 points
5. If question has multiple requirements (both data and explanation), then DUAL path +1 point
6. If question involves simple definitions or basic facts about numbers (e.g., "what is the rate of
return..."), then FACT_ONLY path +4 points
7. If question contains "how", "why", "where", or asks for a process/explanation, then DOC_ONLY path +4 points
8. If question mentions financial terms or ratios and requires interpretation or comparison over time, then
DUAL path +4 points
9. If question includes terms like "ratio", "percentage change", "net change", "average", "percent change", or
similar indicating a need for numerical analysis, then FACT_ONLY path +4 points
```

Figure 8: FINQA Updated Rule

```
1. If question requests numbers, percentages, years, or calculations, especially if it involves comparisons
(e.g., "largest", "fastest"), then FACT_ONLY path +5 points.
2. If question involves historical events or requires detailed context, then DOC_ONLY path +8 points.
3. If question asks for definitions or general knowledge, but does not involve specific data or processes,
then LLM_ONLY path +3 points.
4. If question has multiple requirements (both data and explanation), then DUAL path +10 points.
5. If question includes comparative terms (e.g., "largest", "fastest") but does not explicitly ask for
numbers or specific data, then FACT_ONLY path +5 points, unless it involves detailed context, then DOC_ONLY
path +5 points.
6. If question asks for counts or quantities of items meeting certain criteria (e.g., "how many..."), but
does not require detailed context or historical background, then FACT_ONLY path +5 points.
7. If question asks for a specific fact that can be answered directly from a data source without requiring
additional context or explanation, then FACT_ONLY path +7 points.
8. If question asks for a detailed explanation of a concept or process, then DOC_ONLY path +7 points.
9. If question is about a specific event or process and does not require numerical data, then DOC_ONLY path
+5 points.
```

Figure 9: WIKIQA Updated Rule

```
You are an expert question analyzer. Based on the given question and the following rules, you need to score
each of the four possible answer paths from 0-10.

Question: "{question}"

Rules:
{rules}

Answer Paths:
1. LLM_ONLY: Use general knowledge to answer (for common knowledge questions, definitions, explanations)
2. FACT_ONLY: Search structured data/tables for specific facts, numbers, or data points
3. DOC_ONLY: Search documents/paragraphs for contextual information
4. DUAL: Use both structured data and documents (for complex questions requiring multiple sources)

IMPORTANT: All scores must be integers between 0 and 10 (inclusive). Do not use scores outside this range.

Please analyze the question according to the rules above and provide scores for each path in the following
JSON format:
{{
    "llm_only": <score>,
    "fact_only": <score>,
    "doc_only": <score>,
    "dual": <score>
}}

Only return the JSON object, no other text.
```

Figure 10: Prompt for Rule-Driven Routing Agent

```
You are an expert rule optimizer for a question routing system. Based on the current performance and
recent results, you need to improve the routing rules.

Current Rules:
{current_rules}

Recent Performance Analysis:{Recent Performance Analysis}

Based on this analysis, please suggest improvements to the rules. Consider:
1. Which rules are working well (leading to Good performance) and should be kept/strengthened?
2. Which rules are causing poor performance and should be modified/removed?
3. What new rules should be added based on the patterns you observe?
4. How should rule weights be adjusted to improve Good performance rates?

IMPORTANT:
- All rule scores must be integers between 0 and 10 (inclusive). Do not use scores outside this range.
- Provide ONLY concise rules in the same format as the current rules, with numbered rules and clear
scoring instructions.
- You can ONLY choose from these 4 paths: LLM_ONLY, FACT_ONLY, DOC_ONLY, DUAL.
- Do not add any other text or comments.
```

Figure 11: Prompt for Rule-Driven Routing Agent