

DRAGON: Domain-specific Robust Automatic Data Generation for RAG Optimization

Haiyang Shen^{1,2,*}, Hang Yan^{3,†}, Zhongshi Xing⁴, Mugeng Liu²
Yue Li⁵, Zhiyang Chen^{1,2}, Yuxiang Wang², Jiuzheng Wang², Yun Ma^{1,2,‡}

¹Institute for Artificial Intelligence, Peking University

²School of Computer Science, Peking University

³The Chinese University of Hong Kong

⁴School of Computer Science, Sun Yat-sen University

⁵School of Software & Microelectronics, Peking University

Abstract

Retrieval-augmented generation (RAG) can substantially enhance the performance of LLMs on knowledge-intensive tasks. Various RAG paradigms—including vanilla, planning-based, and iterative RAG—all depend on a robust retriever, yet existing retrievers rely heavily on public knowledge and often falter when faced with domain-specific queries. To address these limitations, we introduce DRAGON, a framework that combines a data-construction modeling approach with a scalable synthetic data-generation pipeline, specifically designed to optimize domain-specific retrieval performance and bolster retriever robustness. To evaluate RAG performance on domain-specific RAGs, we propose DRAGONBENCH, a benchmark spanning 8 domain-specific document collections across 4 distinct fields and featuring a wide spectrum of query complexities, answerability, and hop numbers. Leveraging DRAGON, we generate a large-scale synthetic dataset—encompassing both single-hop and multi-hop queries—to enrich retriever training. Extensive experiments demonstrate that retrievers trained on this data yield significant performance gains and exhibit strong cross-domain generalization. Moreover, when our optimized retrievers are integrated into vanilla, planning-based, and iterative RAG paradigms, we observe consistent end-to-end improvements in system accuracy.

1 Introduction

Retrieval-Augmented Generation (RAG) (Hu and Lu, 2024; Zhao et al., 2024; Gao et al., 2024) can enhance the performance of the large language models (LLM) on knowledge-intensive tasks. Formally, given user queries $\mathcal{Q} = \{q_i\}_{i=1}^m$ and document set $\mathbb{D} = \{D_i\}_{i=1}^N$, the goal of RAG is

to give an answer $\hat{\mathcal{A}}_i$ for each query q_i by referencing \mathbb{D} . The retriever \mathcal{R} selects documents $\mathcal{R}(q_i, \mathbb{D}; \theta) \rightarrow \hat{R}_i$ from \mathbb{D} , and the LLM \mathcal{G} generates an answer, represented as $\mathcal{G}(q_i, \hat{R}_i; \Theta) \rightarrow \hat{\mathcal{A}}_i$.

Existing RAGs (Hu and Lu, 2024) include 3 structures: vanilla, planning-based, and iterative. In vanilla RAG (Guu et al., 2020; Borgeaud et al., 2022), the retriever retrieves once, and the generator gives the answer. In planning-based RAG (He et al., 2023), a planner first devises a strategy composed of multiple sub-queries, followed by the retriever doing retrievals, and the generator synthesizing. In iterative RAG (Yao et al., 2023; Jiang et al., 2024), the LLM interacts iteratively with the retriever, and when the LLM determines that sufficient information has been obtained, it hands over to the generator for synthesis. Regardless of the structure, a robust retriever is crucial for the entire RAG system.

In this paper, we focus on optimizing dense retrievers for domain-specific corpora to boost both retrieval robustness and end-to-end RAG performance. Existing dense retrievers are usually trained on general-purpose sources (e.g., Wikipedia), which hampers their effectiveness on domain-specific content (Lee et al., 2025). This shortfall stems from (1) domain-specific terminology and query styles, (2) semantic ambiguities or some specific constraints that misalign queries and documents, and (3) RAG’s reliance on LLMs to decompose queries into sub-queries of varying completeness, further complicating matching. In specialized domains, authors often omit intermediate assumptions, use diverse and scattered expressions, and require multi-step, cross-paragraph or cross-document reasoning—compounding misalignment and sub-query retrieval challenges. Therefore, a robust retriever must handle a wide range of logical complexities and retrieve documents relevant to both full queries and their partial sub-queries.

In this paper, we aim to enhance the retriever

*Email: hyshen@stu.pku.edu.cn

†Project Leader: hyan@cuhk.edu.hk

‡Corresponding: mayun@pku.edu.cn

used in RAG from the perspective of data synthesis. We propose DRAGON, which incorporates an RAG data synthesis model to capture the complex mapping relationships among documents, queries, ground-truth answers, relevant clues, and their connections to source documents, along with a specific implementation. DRAGON can be used for large-scale RAG data generation for documents in any specific domain, supporting both single-hop and multi-hop queries. It enables the simulation of query generation with varying logical complexities and clue completeness through paraphrasing. Using DRAGON, we synthesize a diverse dataset to enhance the retriever’s performance and the end-to-end performance of the entire RAG system.

To provide the DRAGON with various domain document corpora for synthesizing data and constructing a benchmark to evaluate the domain-specific retriever performance, we collect 8 domain-specific document collections across 4 domains and build DRAGONBENCH. DRAGONBENCH includes queries with varying hop counts. The query may be fully or partially addressed by certain sentences in the corresponding corpus, reflecting situations that frequently occur in real life. It also implements sentence-level citations for the clues supporting the answers and provides detailed evaluation criteria. Building on these evaluation criteria, we developed the criteria-based score generation (CSG) to assess the fidelity of the generator. We find that CSG offers greater stability compared to the basic LLM-as-a-Judge approach (Gu et al., 2025), same as RocketEval (Wei et al., 2025a) and BiGGenBench (Kim et al., 2025) show.

To validate the improvement of DRAGON on retrievers, we select 6 top retrievers from the MTeb leaderboard (Muennighoff et al., 2023) with varying sizes (33~ 611M) and context lengths (512~ 8192). Using contrastive learning (Izacard et al., 2022) and ANCE hard negative sampling (Xiong et al., 2021), we demonstrate significant performance improvement on all 4 large-scale document collections (Hearthstone Wiki Contributors, 2025; Zelda Wiki Contributors, 2025; Drugs.com, 2025; Mayo Clinic, 2025). The ablation study reveals that retriever gains stem not only from in-domain training but also critically from complex logical reformulations and clue-completeness constructions, which substantially boost performance and robustness. Moreover, these enhancements can also generalize to out-of-domain data: a retriever trained on the Zelda (Zelda Wiki

Contributors, 2025) corpus consistently delivers significant improvements on four other domain-specific datasets (Stanford University, 2025; University of California, Berkeley, 2025; Cyotek, 2023; Notion, 2023). Finally, when integrated into various RAG paradigms, our optimized retriever yields stable end-to-end performance gains across the entire system.

In summary, this paper makes the following key contributions:

- We propose DRAGON, a data synthesis method, along with an implementation. DRAGON formally constructs mapping relationships among documents, queries, answers, relevant clues, and their connections to source documents. Based on DRAGON, we introduce a large-scale synthetic dataset characterized by varying logical complexities, clue completeness, and hop counts.
- We introduce DRAGONBENCH, consisting of 8 domain-specific datasets across 4 domains. DRAGONBENCH provides extensive domain coverage, diverse hop coverage, varying levels of question answerability, and finer citation granularity. Utilizing its per-question scoring criteria, we also propose the CSG score, demonstrating its stability over the LLM-as-a-Judge metric.
- Extensive experiments validate DRAGON’s improvements in the retriever across various domains. While existing work primarily focuses on optimizing LLMs themselves, we demonstrate that optimizing the retriever with synthetic data can also enhance the end-to-end consistency performance of the entire RAG system.

2 Related Work

In this Section, we first discuss existing work for RAG (2.1). Second, we introduce a series of optimization approaches utilizing generated data (2.2), particularly related to RAG. Finally, we review existing RAG benchmarks and highlight the distinctions between them and DRAGONBENCH (2.3).

2.1 Retrieval-Augmented Generation

RAG enhances the understanding of queries by incorporating external knowledge. In vanilla RAG (Guu et al., 2020; Borgeaud et al., 2022), the generator and retriever interact once, relying entirely on the retriever’s capabilities. Because the retriever’s intelligence is limited, the RAG’s understanding of user queries depends largely on the

retriever (Zhao et al., 2024; Hu and Lu, 2024). Consequently, recent studies (He et al., 2023; Yao et al., 2023; Jiang et al., 2024; Pang et al., 2024) emphasize leveraging LLMs to enhance query interpretation. Notable approaches include planning (He et al., 2023) and iterative RAG (Yao et al., 2023; Jiang et al., 2024).

Regardless of the structure, a robust retriever is crucial for the entire RAG system. The performance of the RAG system depends not only on the LLM’s ability to paraphrase, decompose, and plan but also heavily relies on the retriever’s adaptability to varying question complexities, answerability, and domain-specific styles.

2.2 Data Synthesis Optimization Methods

The huge success of scaling law (Kaplan et al., 2020) in LLM has spurred data synthesis methods. In NLP (DeepSeek-AI et al., 2025), model distillation from larger LLMs can be used to enhance the capabilities of smaller LLMs. In LLM-based agents (Qin et al., 2024), the integration of LLMs with tools can be optimized using synthetic code (Zhang et al., 2024) and workflow (SHEN et al., 2025; Zeng et al., 2023; Yin et al., 2025). In RAG, data synthesis methods encompass strategies for both the retriever (Ma et al., 2025; Kim and Baek, 2025; Zhou et al., 2024) and the LLM (Devine, 2025; Jadon et al., 2025; Wei et al., 2025b; Chen et al., 2024). For the retriever, data synthesis focuses on generating query-positive-negative triplets using LLMs.

However, existing work still lacks a conceptually clear, domain-specific paradigm for automated data generation and augmentation. In particular, there is a lack of robust studies on enhancing retrievers with respect to the logical complexity and answer completeness of questions, as well as research on their impact on the entire RAG system.

2.3 RAG Benchmarks

RAG benchmarks can be categorized into single, multi-hop, and various hybrid types. These datasets are primarily derived from sources such as Wikipedia, e.g., Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016), PopQA (Mallen et al., 2023), HotpotQA* (Yang et al., 2018), 2WikiMultiHopQA* (Ho et al., 2020), R4C* (Inoue et al., 2020), MuSiQue* (Trivedi et al., 2022), the Freebase knowledge base, e.g., WebQuestions (Berant et al., 2013), or Bing search logs,

e.g., MS MARCO (Nguyen et al., 2016). Hybrid types such as CRAG (Yang et al., 2024) and RAG-Bench (Friel et al., 2025) include a relatively rich set of rag data.

Compared with them, DRAGONBENCH covers 8 domain-specific datasets across 4 areas, featuring varying logical complexity, query answerability, and hops. Additionally, we provided a fine-grained scoring rubric for each question to mitigate the instability issues associated with the naive use of LLM-as-a-Judge for evaluation.

3 DRAGON

3.1 Modeling

Design Rationale. We aim to make the data synthesis broadly applicable across heterogeneous domain corpora (documents, web pages, tables). (1) *Entity graph for multi-hop:* We adopt an entity-centric graph to capture domain-specific terminology, style, and cross-document links in a format-agnostic way. This provides a general mechanism to support single-hop and multi-hop question construction by explicitly bridging sentences across documents via entities and relations. (2) *Rephrasing rules for diversity:* We intentionally keep rephrasing rules simple and randomly initialized to encourage a diverse set of equivalence-preserving transformations. The goal is to cover more domain-specific language phenomena (e.g., terminology, stylistic variation) and maximize query diversities (e.g., metaphor, constraint inclusion).

To unify single and multi-hop, we introduce an intermediate factor to connect different documents, which we refer to as clues. Clues serve as the basis for answering queries and are derived from sentences in one or more documents. For multi-hop data, clues act as bridges linking sentences across documents; for single-hop data, clues correspond to sentences within a single document. Our goal is to construct synthetic data: $\mathcal{G} = (\mathcal{D}, \mathcal{Q}, \mathcal{C}, \mathcal{A}, \mathcal{M})$. The dataset \mathcal{G} consists of:

Document Set (\mathcal{D}) A collection of docs $\mathcal{D} = \{D_i\}_{i=1}^M$, where each D_i is selected from the subset $\mathcal{D} \subseteq \mathbb{D}$ utilized by DRAGON to synthesize data. \mathcal{D} is input and all others are output.

Query Set (\mathcal{Q}) A set of queries $\mathcal{Q} = \{q_i\}_{i=1}^m$, each associated with an answer \mathcal{A}_i in \mathcal{A} .

Clue Set (\mathcal{C}) The set of clues required to answer each query q_i , denoted as $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^m$. Each clue set $\mathcal{C}_i = \{c_{i,j}\}_{j=1}^{u_i}$ consists of clues derived from sentences in documents. These clues serve as inter-

mediate links that connect different documents for addressing multi-hop questions, facilitating query construction.

Answer Set (\mathcal{A}) $\mathcal{A} = \{\mathcal{A}_i^o, \mathcal{A}_{i,j}^v\}_{i=1,j=1}^{m,P_i}$. For convenience, we use \mathcal{A}_i to uniformly represent both \mathcal{A}_i^o and \mathcal{A}_i^v with specific combination of retrieved documents.

- $\mathcal{A}^o = \{\mathcal{A}_i^o\}_{i=1}^m$ comprises standard answers, representing the complete response when all necessary documents are retrieved. Each \mathcal{A}_i^o consists of multiple sentences, $\mathcal{A}_i^o = \{s_{i,j}\}_{j=1}^{n_i}$, where each sentence may include a citation to the original text, as shown in Equation 1: $(D_{i,j,k}, S_{i,j,k,l})$. In the actual data generation, we can first add a reference to the clue after each $s_{i,j}$ as \mathcal{M}_1 shows, and then map the clue \mathcal{C}_i to get $(D_{i,j,k}, S_{i,j,k,l})$ through \mathcal{M}_2 .
- For each \mathcal{A}_i^o , there exists a set of answer variants $\{\mathcal{A}_{i,j}^v\}_{j=1}^{P_i}$, where $P_i = \sum_{k=1}^{p_i} \binom{k}{p_i}$ and p_i is the number of documents (e.g., hops). These variants $\mathcal{A}_{i,j}^v$ capture alternative forms of \mathcal{A}_i^o when one or more supporting documents are missing from retrieval.

Mapping Relationships (\mathcal{M})

- **Answer-to-Clue Mapping (\mathcal{M}_1)** This mapping associates sentence $s_{i,j}$ in the answer \mathcal{A}_i with one or more corresponding clues.
- **Clue-to-Document Mapping (\mathcal{M}_2)** This mapping links each clue $c_{i,j}$ to the specific document-sentence pairs that support it:

$$\begin{aligned} \mathcal{M}_2^D(c_{i,j}) &= \{D_{i,j,k}\}_{k=1}^{r_{i,j}}, \\ \mathcal{M}_2^S(c_{i,j}) &= \left\{ \left\{ (D_{i,j,k}, S_{i,j,k,l}) \right\}_{l=1}^{y_{i,j,k}} \right\}_{k=1}^{r_{i,j}} \end{aligned} \quad (1)$$

where $D_{i,j,k}$ is the k -th document associated with clue $c_{i,j}$, $S_{i,j,k,l}$ is the specific sentence within $D_{i,j,k}$ that provides the information to answer query q_i , $y_{i,j,k}$ is the number of key sentences required from $D_{i,j,k}$ to answer q_i , and $r_{i,j}$ is the number of associated documents with clue $c_{i,j}$.

Using the mappings \mathcal{M}_1 and \mathcal{M}_2 , for each query $q_i \in \mathcal{Q}$, we can identify the corresponding document set $\mathcal{D}(q_i)$ and the sentences $\mathcal{S}(q_i)$ required to answer the query. Specifically, the answer \mathcal{A}_i for q_i is first mapped to its set of clues \mathcal{C}_i via the answer-to-clue mapping $\mathcal{M}_1(\mathcal{A}_i)$. Subsequently, the document set $\mathcal{D}(q_i)$ is defined as the union of

all documents associated with each clue $c_{i,j} \in \mathcal{C}_i$ through the clue-to-document mapping \mathcal{M}_2 :

$$\mathcal{D}(q_i) = \bigcup_{c_{i,j} \in \mathcal{M}_1(\mathcal{A}_i)} \{D \mid D \in \mathcal{M}_2^D(c_{i,j})\} \quad (2)$$

where $|\mathcal{D}(q_i)| = p_i$. p_i denotes the number of required documents. The set of specific sentences $\mathcal{S}(q_i)$ required to answer q_i is obtained by aggregating all supporting sentences from the mapped documents for each clue in \mathcal{C}_i :

$$\mathcal{S}(q_i) = \bigcup_{c_{i,j} \in \mathcal{M}_1(\mathcal{A}_i)} \{S_{i,j,k,l} \mid (D_{i,j,k}, S_{i,j,k,l}) \in \mathcal{M}_2^S(c_{i,j})\} \quad (3)$$

where $|\mathcal{S}(q_i)| = w_i$. w_i represents the total supporting sentences for q_i . To optimize the RAG system, both the retriever and generator components are trained using the synthetic dataset \mathcal{G} .

For the retriever, the required synthetic data consists of $\{\mathcal{D}, \mathcal{Q}\}$, and the mapping from \mathcal{Q} to \mathcal{D} derived from \mathcal{M} . For the entire RAG, the complete structure $\mathcal{G} = (\mathcal{D}, \mathcal{Q}, \mathcal{C}, \mathcal{A}, \mathcal{M})$ is necessary. Additionally, it is essential to construct actual retrieval results using a real retriever based on \mathcal{G} , as the document set provided to the generator during real RAG operation may not fully answer the query. Therefore, we need to consider generating corresponding $\mathcal{A}_{i,j}^v$ under various retrieval scenarios.

3.2 Implementation

Figure 1 illustrates an implementation of DRAGON. The process begins by partitioning documents (①) into multiple chunks, thereby forming the document set \mathcal{D} (②). We chunk documents to align retrieval granularity with the retriever context length. From \mathcal{D} , factual clues are extracted. For single-hop queries, clues are directly extracted (③). In the case of multi-hop queries, connections between multiple documents are established by entities extracted from clues to facilitate question construction (④). After performing entity resolution (⑤), an entity graph is constructed (⑥). Subsequently, the extracted clues are input into an LLM to generate straightforward questions (⑦). For multi-hop questions, multiple entity-centric clues are provided to the LLM to generate more complex, entity-related queries (⑦). Following this, based on predefined equivalence transformation rules (⑨), the LLM is prompted to iteratively reformulate the generated questions (⑧) and vary the completeness of the

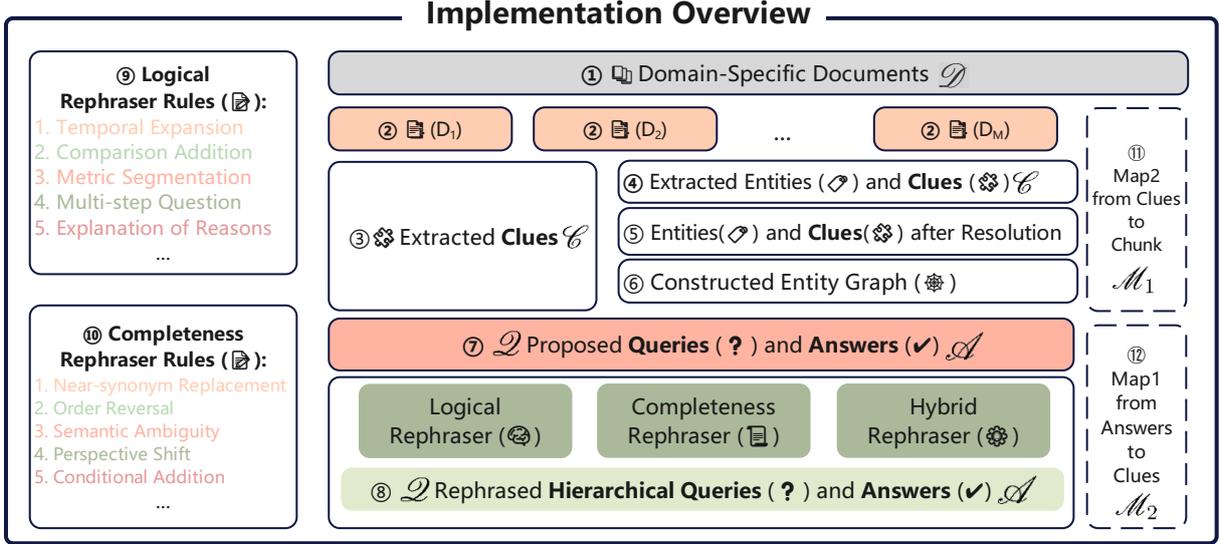


Figure 1: A specific implementation of the DRAGON.

clues (⑩). This results in the construction of data with multiple levels of comprehension difficulty and varying degrees of clue completeness.

4 Document Corpus and DRAGONBENCH

To provide the DRAGON with a corpus for synthesizing and constructing a benchmark, we curate 8 domain-specific document corpora across 4 domains and build DRAGONBENCH (4.1). We introduce the construction process (4.2). Then we introduce the evaluation metrics to assess the performance of the retriever and the generator (4.3).

4.1 Domain-specific Corpus

The principle for data selection was to cover a broad range of applications with high practical value. Thus, we chose datasets from the following areas: (1) game guides (Hearthstone (Hearthstone Wiki Contributors, 2025) and Zelda (Zelda Wiki Contributors, 2025)); (2) medical guides (Drug (Drugs.com, 2025) and Mayo Clinic (Mayo Clinic, 2025)); (3) university admissions (Stanford (Stanford University, 2025) and Berkeley (University of California, Berkeley, 2025)); and (4) software documentation (Cytotec (Cytotec, 2023) and Notion (Notion, 2023)). These datasets were thoroughly cleaned using an LLM. For detailed cleaning procedures, please refer to Appendix A.1. Our corpus selection follows two principles: *diversity across domains* and *public accessibility* of source documents. Details on annotator qualifications and selection are provided in Appendix A.5.

4.2 DRAGONBENCH Construction

We first selected sub-documents and then segmented these documents into chunks. We then built an entity–relationship graph from these chunks and, based on its connections, assigned related chunks to domain experts (e.g., Zelda and Hearthstone enthusiasts) to generate single-hop and multi-hop datasets. Once questions were generated and answered, we asked annotators to reformulate them according to their personal preferences, thereby producing a rich collection of questions, answers, and their source documents and sentences. Building on this, we added question-quality assessment and answer-verification steps to filter the items. Finally, we used an LLM to draft scoring rubrics for each question and had the original question authors conduct two rounds of review—only questions whose rubrics passed both reviews were retained. We finally get DRAGONBENCH. Importantly, the construction of the evaluation set is independent of our synthesized training data, which helps ensure fair assessment of retriever and RAG performance. Annotator recruitment criteria and qualifications are summarized in Appendix A.5. The final dataset is presented in Table 1. For detailed construction procedures and examples, please refer to the Appendix A.2.

4.3 Task Metrics

Retrieval To align with the RAG system’s requirements, we set each retrieval to return $k \in \{3\}$ chunks. The RAG system can be iteratively retrieved multiple times to gather the in-

Table 1: Basic information of DRAGONBENCH. The Games and Mediguides corpus is relatively large, making it suitable for synthetic data, while the Universities and Documentation corpus is comparatively small, ideal for validating cross-domain generalization.

DRAGONBENCH	Games		Mediguides		Universities		Documentation	
	Hearthstone	Zelda	Drug	Mayoclinic	Stanford	Berkeley	Cyotek	Notion
# Docs	15695	10665	77348	2542	65	280	808	475
# Avg Tokens	1112	746	2159	1369	1052	1283	478	4382
# Single Hop	873	870	826	862	272	860	653	877
# Multi Hop	162	390	183	234	162	398	259	447

Table 2: The score inconsistency between different model on Zelda. The consistency of CSG between models and humans significantly outperforms the LJ

	LJ	CSG
Qwen2-72B	56.39%	52.03%
Qwen2.5-72B	62.18%	54.51%
Qwen2-72B vs Qwen2.5-72B	5.79%	2.48%
Llama3-70B	62.18%	59.55%
Llama3.1-70B	53.70%	54.20%
LLama3-70B vs Llama3.1-70B	8.48%	5.35%
Qwen2.5-72B vs Llama3.1-70B	8.48%	0.31%
Inconsistency with Human	24.35%	8.68%

formation deemed necessary by the LLM. Therefore, we assess retrieval using $\text{Precision}@k^{\text{avg}} = \frac{1}{m} \left(\sum_{i=1}^m \frac{|\{D \in \mathcal{D}^k(q_i) | D \in \hat{\mathcal{D}}(q_i)\}|}{k} \right)$, and $\mathcal{D}^k(q_i)$ denotes the top k documents retrieved by the retriever.

Entire RAG system To evaluate the overall RAG system, existing metrics include the Exact Match (EM) and LLM-as-a-Judge (LJ) (Gu et al., 2025). The EM can provide stable evaluations but lacks robustness for complex responses and is sensitive to formatting variations. The LJ leverages LLM to assess the alignment between generated and reference answers, thereby mitigating some limitations of EM, but it can suffer from evaluation instability. To address these limitations, we introduce the *Criteria-based Score for Generation* (CSG), $s \equiv_{\text{LLM}} \hat{\mathcal{A}}_i$ means the LLM confirms that s matches $\hat{\mathcal{A}}_i$:

$$\text{CSG} = \frac{1}{m} \sum_{i=1}^m \frac{|\{s \in \mathcal{S}(q_i) | s \equiv_{\text{LLM}} \hat{\mathcal{A}}_i\}|}{|w_i|}. \quad (4)$$

In the formula, m is the number of queries; for each q_i , $\mathcal{S}(q_i)$ is the set of supporting sentences derived via \mathcal{M}_1 and \mathcal{M}_2 , and $|w_i| \equiv |\mathcal{S}(q_i)|$. The predicate $s \equiv_{\text{LLM}} \hat{\mathcal{A}}_i$ denotes the LLM’s criterion-aligned judgment that sentence s satisfies the per-question rubric associated with $\hat{\mathcal{A}}_i$. We provide

complete variable definitions in Appendix A.6.

From Table 2, we observe that the consistency of CSG is superior. This is primarily because the detailed criteria customized for each question in CSG can mitigate the problem of instability associated with using LLM-as-a-Judge alone. Similar conclusions have also been reached by other work, such as (Wei et al., 2025a) and (Kim et al., 2025).

5 Evaluation

5.1 Setup

We use Qwen2.5-72B-Instruct. When generating data, we randomly sampled a *subset* of corpus documents for synthesis (see Table 7). Empirically, this limited synthesis was sufficient to yield substantial gains in domain-specific retrieval, as evidenced by our main and ablation results. A preliminary breakdown of resource usage is provided in Appendix A.7. Additional details of the synthesized training data are summarized in Appendix A.3.

Main Experiments: The main experiment involves fine-tuning the retriever with synthetic data to enhance the performance and robustness of the retriever.

Based on the MTeb leaderboard (Muennighoff et al., 2023), we chose 6 models of varying sizes and contexts that exhibited the best retrieval performance. The selected retrieval models, with their size and context length (CL): stella_en_400M_v5 (Zhang et al., 2025) (435M, CL 8192), gte-multilingual-base (Alibaba-NLP) (611MB, CL 8192), snowflake-arctic-embed-m-long/v1.5 (Merrick, 2024), (547M, CL 2048; 109M, CL 512), rubert-tiny-turbo (Sergeyzh) (117M, CL 2048), and MedEmbed-small-v0.1 (Krishnappa, 2025) (33M, CL 512). We optimize the retriever using contrastive loss (Izacard et al., 2022) and ANCE sampling (Xiong et al., 2021). For convenience, we used the same hyperparameters across all experiments. Due to limitations in

Table 3: Improvements to the retriever. Blue text indicates performance improvement. ORI represents the retrieval performance of fully answerable queries on the unoptimized retriever, BEF denotes performance on the unoptimized retriever when data has partial clue completeness. As shown, BEF underperforms compared to ORI, indicating that the existing retriever struggles with domain-specific corpora. After optimization, all retrievers achieve consistent performance improvements across all datasets.

Precision@3 (% / 100)	Games								Mediguides								
	Hearthstone				Zelda				Drug				Mayoclinic				
	ORI	BEF	AFT	↑ (%)	ORI	BEF	AFT	↑ (%)	ORI	BEF	AFT	↑ (%)	ORI	BEF	AFT	↑ (%)	
stella_en_400M_v5 (Zhang et al., 2025)	SH	26.78	20.94	32.22	↑ 53.87	34.48	30.68	49.88	↑ 62.58	69.49	62.61	72.24	↑ 15.38	49.07	49.45	60.17	↑ 21.68
	MH	13.55	12.38	23.98	↑ 93.70	11.11	8.33	25.97	↑ 211.8	32.97	22.09	46.51	↑ 110.6	31.98	30.33	37.71	↑ 24.33
gte-base-multilingual (Alibaba-NLP)	SH	34.02	25.15	41.26	↑ 64.06	43.18	36.55	51.96	↑ 42.16	65.38	50.70	71.48	↑ 40.99	56.15	56.03	68.33	↑ 21.95
	MH	24.66	23.73	33.67	↑ 41.89	20.27	22.34	30.56	↑ 36.79	36.98	18.22	53.35	↑ 192.8	37.32	34.83	46.62	↑ 33.85
snowflake-arctic-m-long (Merrick, 2024)	SH	8.74	9.54	36.43	↑ 281.9	13.29	15.65	44.58	↑ 184.9	32.32	18.88	75.51	↑ 300.0	24.59	20.58	68.57	↑ 233.2
	MH	7.31	9.36	36.50	↑ 290.0	6.58	9.42	26.60	↑ 182.4	15.03	10.17	56.22	↑ 452.8	16.10	13.89	48.57	↑ 249.7
snowflake-arctic-v1.5 (Merrick, 2024)	SH	9.08	12.39	33.21	↑ 168.0	11.80	19.56	46.09	↑ 135.6	43.95	54.12	70.34	↑ 29.97	21.11	36.18	66.99	↑ 85.16
	MH	9.32	14.67	36.21	↑ 146.8	4.53	9.42	31.52	↑ 234.6	35.97	39.05	45.54	↑ 16.62	25.36	31.98	47.82	↑ 49.53
rubert-tiny-turbo (Sergeyzh)	SH	0.23	0.00	9.17	↑ ∞	2.41	1.22	10.64	↑ 772.1	7.87	3.30	19.26	↑ 483.6	12.99	9.01	22.53	↑ 150.1
	MH	0.00	0.00	8.67	↑ ∞	0.00	0.00	8.57	↑ ∞	1.82	1.65	13.18	↑ 698.8	10.68	8.11	21.65	↑ 167.0
MedEmbed-small-v0.1 (Krishnappa, 2025)	SH	28.05	20.94	29.62	↑ 41.45	37.92	27.75	43.30	↑ 56.04	65.74	55.51	71.61	↑ 29.00	56.84	53.59	63.82	↑ 19.09
	MH	26.28	23.73	32.12	↑ 35.36	19.55	14.86	31.40	↑ 111.3	46.90	35.85	54.77	↑ 52.78	39.32	38.14	47.98	↑ 25.80

computational resources, we used LoRA with a relatively big rank to train uniformly for 10,000 steps with a learning rate of $2e-5$. For detailed training parameters, please refer to the Appendix A.4.

Generalization Experiments: We evaluated the generalization performance of the trained models on the remaining 4 corpora.

Overall RAG System Improvement Experiments: For the entire RAG system, users care whether it can effectively solve their queries. We believe that optimizing the retriever can enhance the overall performance. We select 4 baselines: Vanilla (Guu et al., 2020; Borgeaud et al., 2022), RR (Retrieve and Rank) (He et al., 2023), where 3 results are retained per retrieval and 5 results are kept for generation after aggregating; Flare Style RAG (Jiang et al., 2024), which initially generates a sentence using a planner, then uses this sentence for retrieval to verify its accuracy, iterating up to a maximum of 8 times; and ReACT (Yao et al., 2023), where the planner independently decides when to retrieve, aggregate, and stop, with a maximum of 8 interactions. The base model in use remains Qwen2.5-72B-Instruct.

5.2 Main Experiments

Table 3 demonstrates the performance improvements. It can be observed that existing retrievers perform poorly when faced with domain-specific queries. However, their performance can be significantly improved after optimizing with DRAGON. For convenience, we used the same hyperparameters across all experiments. As a result, models may not achieve optimal performance, indicating room

for improvement. Additionally, we can clearly see that some retrievers trained on general data perform very poorly. This further underscores the importance of an automated RAG data synthesis framework tailored to specific domains for enhancing retriever performance.

5.3 Ablation Experiments

Table 4 presents the performance of our various data augmentation methods, specifically logical transformations and completeness transformations, across different domain datasets. It is evident that the new data synthesized through logical and completeness transformations significantly enhances the retriever’s effectiveness on domain-specific data. This approach facilitates a deeper understanding of domain-specific datasets by the retriever.

More ablation study results about the retriever are reported in Appendix A.8.

5.4 Generalization Experiments

Tables 5 present the performance of retrievers when fine-tuned on Zelda and applied to other datasets. It is evident that the data fine-tuned with DRAGON generalizes well to out-of-domain tasks for retrievers. This implies that logical transformations and completeness transformations are not only effective for training in specific domains but also possess a certain degree of generalization capability for out-of-domain problems.

5.5 Overall RAG System Experiments

Tables 6 show the performance improvements across the entire RAG system when using a fine-

Table 4: Through continuous transformation of original questions, additional improvements can be achieved beyond those gained from domain-specific fine-tuning.

Precision@3 (% / 100)		Games				Mediguides			
		Hearthstone		Zelda		Drug		Mayoclinic	
		SH	MH	SH	MH	SH	MH	SH	MH
stella_en_400M_v5 (Zhang et al., 2025)	w/o Logical Transformation	28.32	20.18	45.23	22.24	65.23	41.98	55.32	30.21
	w/o Completeness Transformation	25.54	18.23	43.33	21.45	64.32	39.32	52.76	32.22
	Full	32.22	23.98	49.88	25.97	72.24	46.51	60.17	37.71
snowflake-arctic-v1.5 (Merrick, 2024)	w/o Logical Transformation	28.28	34.32	43.78	28.75	64.32	38.32	60.21	44.87
	w/o Completeness Transformation	26.21	30.21	41.76	25.69	62.23	40.22	57.43	43.76
	Full	33.21	36.21	46.09	31.52	70.34	45.54	66.94	47.82
MedEmbed-small-v0.1 (Krishnappa, 2025)	w/o Logical Transformation	28.32	32.56	41.23	26.43	63.42	47.32	58.34	42.23
	w/o Completeness Transformation	25.38	30.34	37.24	24.24	62.56	50.35	53.43	45.34
	Full	29.62	32.12	43.30	31.40	71.61	54.77	63.82	47.98

Table 5: The retriever fine-tuned on Zelda consistently enhances the retrievers on other domains.

Precision@3 (% / 100)		Universities				Documentation				Games		Mediguides			
		Stanford		Berkeley		Cytok		Notion		Hearthstone		Drug		Mayoclinic	
		SH	MH	SH	MH	SH	MH	SH	MH	SH	MH	SH	MH	SH	MH
stella_en_400M_v5 (Zhang et al., 2025)	BEF	59.52	27.05	37.96	15.79	47.80	25.72	13.77	2.21	20.94	12.38	62.61	22.09	49.45	30.33
	AFT	61.90	31.72	39.19	16.52	51.22	29.31	14.49	3.39	25.40	18.11	63.24	25.48	52.98	32.13
	↑ (%)	↑ 3.40	↑ 14.72	↑ 3.24	↑ 4.62	↑ 7.15	↑ 13.96	↑ 5.23	↑ 53.39	↑ 21.30	↑ 46.28	↑ 1.01	↑ 15.35	↑ 7.14	↑ 5.93
snowflake-arctic-v1.5 (Merrick, 2024)	BEF	48.41	24.77	26.04	10.22	40.33	26.44	11.74	3.98	12.39	14.67	54.12	39.05	36.18	31.98
	AFT	59.13	28.54	39.93	13.20	51.38	30.53	18.20	4.26	24.41	24.61	66.79	44.96	56.64	39.79
	↑ (%)	↑ 22.14	↑ 15.22	↑ 53.34	↑ 29.16	↑ 27.40	↑ 15.47	↑ 55.03	↑ 7.04	↑ 97.01	↑ 67.76	↑ 23.41	↑ 15.13	↑ 56.55	↑ 24.42
MedEmbed-small-v0.1 (Krishnappa, 2025)	BEF	55.56	23.40	35.26	12.81	43.90	25.22	12.69	2.36	20.94	23.73	55.51	35.85	53.59	38.14
	AFT	58.73	28.77	37.97	14.80	48.13	32.18	15.69	3.78	29.24	28.51	67.17	43.80	57.13	41.22
	↑ (%)	↑ 5.71	↑ 22.95	↑ 7.69	↑ 15.53	↑ 9.64	↑ 27.60	↑ 23.64	↑ 60.17	↑ 39.64	↑ 20.14	↑ 21.01	↑ 22.18	↑ 6.61	↑ 8.08

Table 6: The DRAGON-optimized retriever consistently enhances the performance of both single-hop and multi-hop RAG across various methods. This improvement primarily stems from the retriever’s enhanced understanding of the fine-grained semantics in user queries, enabling the LLM to obtain more robust retrieval results with fewer interaction steps through more casual sub-queries. SH stands for Single Hop, and MH stands for Multi Hop.

Precision@3 (% / 100)		ReACT Style (Yao et al., 2023)				vanillaRAG (Gao et al., 2024)		Flare Style (Jiang et al., 2024)		RR Style (He et al., 2023)	
		4-Iter		8-Iter		SH	MH	SH	MH	SH	MH
		SH	MH	SH	MH						
stella_en_400M_v5 (Zhang et al., 2025)	BEF	51.52	39.43	39.05	53.22	49.00	43.88	51.86	57.40	57.62	53.79
	AFT	52.45	43.32	43.23	55.23	53.24	45.72	55.34	60.23	62.34	56.43
	↑ (%)	↑ 1.81	↑ 9.87	↑ 10.93	↑ 3.78	↑ 8.65	↑ 4.19	↑ 6.71	↑ 4.93	↑ 8.19	↑ 4.91
snowflake-arctic-v1.5 (Merrick, 2024)	BEF	45.19	39.88	38.19	44.99	37.05	34.76	50.12	59.48	53.40	50.57
	AFT	46.95	42.38	42.19	47.17	40.21	41.85	55.23	58.14	57.57	55.25
	↑ (%)	↑ 3.89	↑ 6.27	↑ 10.47	↑ 4.85	↑ 8.53	↑ 20.40	↑ 10.20	↓ 2.25	↑ 7.81	↑ 9.25
MedEmbed-small-v0.1 (Krishnappa, 2025)	BEF	45.47	40.41	44.92	39.94	40.33	32.54	46.80	58.43	50.73	53.40
	AFT	45.66	44.95	44.05	44.35	43.16	41.62	54.95	66.77	54.11	52.84
	↑ (%)	↑ 0.42	↑ 0.42	↓ 1.94	↑ 11.04	↑ 7.02	↑ 27.90	↑ 17.41	↑ 14.27	↑ 6.63	↓ 1.05

tuned retriever. It is evident that a robust retriever can lead to steady enhancements in the overall performance of the RAG system. This improvement may primarily stems from the enhanced robustness of the retriever to queries, achieved through training with completeness data, which develops robust adaptability to various queries and subqueries. Our retriever not only retrieves queries and documents

with direct semantic correspondence more accurately but also demonstrates strong robustness in cases where queries and documents are partially misaligned.

6 Conclusion

In this work, we introduced DRAGON, a framework for synthesizing RAG data that captures the complex relationships between queries, documents,

answers, and supporting clues. DRAGON enables the generation of diverse, domain-specific datasets with varying logical complexities and clue completeness, enhancing the robustness of retrievers. We also develop DRAGONBENCH, a benchmark spanning 8 document corpora across 4 domains. The extensive experiments demonstrated significant improvements in retriever performance across multiple domains and the RAG paradigm. These results validate the effectiveness and generalizability of DRAGON.

7 Limitations

The types of synthesis and evaluation questions are limited. The existing implementation of DRAGON concentrates on generating fact-based QA data with deterministic answers, rather than tackling more complex questions that involve logical reasoning across different entities within documents. This issue is not unique to our work; current research also focuses on evaluating "factual questions" for ease of assessment. Further exploration of open-ended questions remains necessary.

Assessing the Authenticity of Benchmark Questions. We cannot guarantee that the questions we construct reflect those that would be genuinely asked by humans in the real world. We believe this is something that cannot be ensured for most existing RAG evaluations, as their QA datasets are rarely collected from real-world scenarios. Instead, they are often crowdsourced or generated by language models. However, we do not see this as entirely necessary, as we have conducted generalization experiments that demonstrate the model's ability to generalize to some extent.

Limitations of Domain Selection. Our research focuses on domain-specific retrieval optimization up to RAG system optimization. We have made efforts to collect publicly available domain-specific datasets that we consider valuable and have used LLMs for data cleansing. However, in the real world, domain-specific data is more complex, and we cannot cover all possible scenarios.

References

Alibaba-NLP. Gte multilingual base. <https://huggingface.co/Alibaba-NLP/gte-multilingual-base>.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. [Semantic parsing on Freebase from question-answer pairs](#). In *Proceedings of the 2013*

Conference on Empirical Methods in Natural Language Processing, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Mingda Chen, Xilun Chen, and Wen-tau Yih. 2024. [Few-shot data synthesis for open domain multi-hop question answering](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 190–208, St. Julian's, Malta. Association for Computational Linguistics.

Cyotek. 2023. Cyotek. <https://www.cyotek.com/>.

DeepSeek-AI, Daya Guo, Dejian Yang, and etc. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.

Peter Devine. 2025. [Alofrag: Automatic local fine tuning for retrieval augmented generation](#). *Preprint*, arXiv:2501.11929.

Drugs.com. 2025. Drugs.com. <https://www.drugs.com/>.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2025. [Ragbench: Explainable benchmark for retrieval-augmented generation systems](#). *Preprint*, arXiv:2407.11005.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#). *Preprint*, arXiv:2312.10997.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. [A survey on llm-as-a-judge](#). *Preprint*, arXiv:2411.15594.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Realm: retrieval-augmented language model pre-training](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org.

- Hangfeng He, Hongming Zhang, and Dan Roth. 2023. [Rethinking with retrieval: Faithful large language model inference](#). *Preprint*, arXiv:2301.00303.
- Hearthstone Wiki Contributors. 2025. [Hearthstone wiki](https://hearthstone.fandom.com/wiki/Hearthstone_Wiki). https://hearthstone.fandom.com/wiki/Hearthstone_Wiki.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Yucheng Hu and Yuxing Lu. 2024. [Rag and rau: A survey on retrieval-augmented language model in natural language processing](#). *arXiv preprint arXiv:2404.19543*.
- Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. [R4C: A benchmark for evaluating RC systems to get the right answer for the right reason](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750, Online. Association for Computational Linguistics.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised dense information retrieval with contrastive learning](#). *Transactions on Machine Learning Research*.
- Aryan Jadon, Avinash Patil, and Shashank Kumar. 2025. [Enhancing domain-specific retrieval-augmented generation: Synthetic data generation and evaluation using reasoning models](#). *Preprint*, arXiv:2502.15854.
- Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2024. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Minsang Kim and Seung Jun Baek. 2025. [Syntriever: How to train your retriever with synthetic data from LLMs](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2523–2539, Albuquerque, New Mexico. Association for Computational Linguistics.
- Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2025. [The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5877–5919, Albuquerque, New Mexico. Association for Computational Linguistics.
- Abhinand Krishnappa. 2025. [Medembed](https://github.com/abhinand5/MedEmbed). <https://github.com/abhinand5/MedEmbed>. Accessed: 2025-02-12.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. [NV-embed: Improved techniques for training LLMs as generalist embedding models](#). In *The Thirteenth International Conference on Learning Representations*.
- Xueguang Ma, Xi Victoria Lin, Barlas Oguz, Jimmy Lin, Wen tau Yih, and Xilun Chen. 2025. [Drama: Diverse augmentation from large language models to smaller dense retrievers](#). *Preprint*, arXiv:2502.18460.
- Alex Mullen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.
- Mayo Clinic. 2025. [Mayo clinic](https://www.mayoclinic.org/). <https://www.mayoclinic.org/>.
- Luke Merrick. 2024. [Embedding and clustering your data can improve contrastive pretraining](#). *Preprint*, arXiv:2407.18887.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. [MTEB: Massive text embedding benchmark](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. **MS MARCO: A human generated machine reading comprehension dataset**. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Notion. 2023. Notion help center. <https://www.notion.com/help>.
- Jing-Cheng Pang, Heng-Bo Fan, Pengyuan Wang, Jiahao Xiao, Nan Tang, Si-Hang Yang, Chengxing Jia, Sheng-Jun Huang, and Yang Yu. 2024. Empowering language models with active inquiry for deeper understanding. *arXiv preprint arXiv:2402.03719*.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. **ToolLLM: Facilitating large language models to master 16000+ real-world APIs**. In *The Twelfth International Conference on Learning Representations*.
- 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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Sergey Zh. Rubert-tiny-turbo. <https://huggingface.co/sergeyzh/rubert-tiny-turbo>.
- Haiyang SHEN, Yue Li, Desong Meng, Dongqi Cai, Sheng Qi, Li Zhang, Mengwei Xu, and Yun Ma. 2025. **Shortcutsbench: A large-scale real-world benchmark for API-based agents**. In *The Thirteenth International Conference on Learning Representations*.
- Stanford University. 2025. Stanford university admissions. <https://admission.stanford.edu/>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. **MuSiQue: Multi-hop questions via single-hop question composition**. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- University of California, Berkeley. 2025. University of California, Berkeley admissions. <https://admissions.berkeley.edu/>.
- Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. 2025a. **Rocketeval: Efficient automated LLM evaluation via grading checklist**. In *The Thirteenth International Conference on Learning Representations*.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025b. **Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales**. *Preprint*, arXiv:2406.13629.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. **Approximate nearest neighbor negative contrastive learning for dense text retrieval**. In *International Conference on Learning Representations*.
- Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu JIANG, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas SCHEFFER, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. **CRAG - comprehensive RAG benchmark**. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. **HotpotQA: A dataset for diverse, explainable multi-hop question answering**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- 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*.
- Fan Yin, Zifeng Wang, I-Hung Hsu, Jun Yan, Ke Jiang, Yanfei Chen, Jindong Gu, Long T. Le, Kai-Wei Chang, Chen-Yu Lee, Hamid Palangi, and Tomas Pfister. 2025. **Magnet: Multi-turn tool-use data synthesis and distillation via graph translation**. *Preprint*, arXiv:2503.07826.
- Zelda Wiki Contributors. 2025. Zelda wiki. https://zelda.fandom.com/wiki/Main_Page.
- Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. **Agenttuning: Enabling generalized agent abilities for llms**. *Preprint*, arXiv:2310.12823.
- Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. **Jasper and stella: distillation of sota embedding models**. *Preprint*, arXiv:2412.19048.
- Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, Zhiwei Liu, Yihao Feng, Tulika Awalgaonkar, Rithesh Murthy, Eric Hu, Zeyuan Chen, Ran Xu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. 2024. **xlam: A family of large action models to empower ai agent systems**. *Preprint*, arXiv:2409.03215.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Junjie Zhou, Zheng Liu, Ze Liu, Shitao Xiao, Yueze Wang, Bo Zhao, Chen Jason Zhang, Defu Lian, and Yongping Xiong. 2024. *Megapairs: Massive data synthesis for universal multimodal retrieval*. *Preprint*, arXiv:2412.14475.

A Appendix

A.1 Detailed Source Data Cleaning Procedures

The source data we obtained is from web pages. Therefore, we conducted the following data cleaning processes: extracting the main HTML content to eliminate irrelevant information, redirecting hyperlinks to convert them into local links, and converting files of different formats, such as extracting the main content from HTML and converting PDFs to text. Subsequently, we processed the resulting files using Qwen2.5-72B-Instruct to further remove any potentially invalid information from the original files. A cleaned sample prompt is as follows:

System & User Prompts

System Prompt

You are an assistant tasked with cleaning and formatting text converted from HTML or PDF documents. The goal is to remove unnecessary content like formatting errors, irrelevant metadata, conversion artifacts, advertisements, logos, login pages, navigation menus, footers, social media buttons, copyright notices, terms and conditions, empty lists, incomplete sections, and other pages lacking useful content, while keeping the text true to the original meaning. Do not change the meaning of any part of the document. Ensure the output follows clean Markdown formatting.

User Prompt

Your task is to: 1. Remove all irrelevant or incomplete content mentioned. 2. Correct any formatting issues and output the text in proper Markdown format. 3. Keep the original meaning and content as accurate as possible.

Here is the text:

```
{content_chunk}
```

A.2 Detailed Bench Construction Procedures and Examples

Our goal is to enhance retrieval and overall RAG system performance on domain-specific datasets. To achieve this, we need an evaluation benchmark to assess retrieval and RAG performance on these

datasets. We established this benchmark using the following methods: (1) Single-hop Questions: After segmenting selected documents, each segment is randomly assigned to a domain-specific expert (e.g., Zelda enthusiasts or Hearthstone fans). These experts generate 3-6 questions per document segment using their perspective, assisted by LLMs or Google; (2) Multi-hop Questions: We leveraged the entity relationship graph component from DRAGON to link multiple documents. Annotators were provided with documents connected by the same entity or relationship and tasked with generating 3-6 questions based on these documents. The corresponding answer sentences and answers were also recorded. All annotations are the author.

The questions generated from these steps are typically direct Q&A questions targeting entities or relationships. To simulate the variety of questions people might ask in real scenarios, we further enhanced each question by having different individuals modify them in various ways, such as introducing ambiguity, adding constraints, or including unanswerable yet related parts.

Question Generation Instructions

- Inputs
 - Single-hop: one document segment (1-2 paragraphs on a single topic)
 - Multi-hop: a small set (2-4) of segments linked by shared entities or relations
- For each input, write 3-6 questions:
 - Single-hop questions must be answerable by reading only that one segment.
 - Multi-hop questions must require combining facts from all segments in the set.
- Question style
 - Focus on factual queries (Who/What/When/Where/How many/How).
 - Keep each question clear and unambiguous.
 - Do not introduce information not present in the segment(s).
- Answer annotation
 - For each question, provide:
 - The exact sentence(s) from the segment(s) containing the answer (extractive span).
 - The concise answer text.

Finally, all generated questions were filtered using LLMs based on metrics like practical value, resulting in our final set of questions. Subsequently, we developed detailed scoring criteria for each problem, which were then verified manually.

The compensation for generating each question is based on local pricing, that is 1 RMB per question in China, with additional payments for each enhanced variation.

A sample question is as follows:

A Question example

How many Side Quests are there in total in The Legend of Zelda: Breath of the Wild and its DLC Packs?

A transformed sample is as follows:

A Transformed Example

How many Side Quests were added to The Legend of Zelda: Breath of the Wild and its DLC Packs, and how does this number compare to the number of Side Quests in The Witcher 3: Wild Hunt?

An example of a scoring criterion is as follows:

Scoring Criterion Example

(1) Award 1 point if the response indicates that the respondent provides the correct quantity.
(2) Award 1 point if the response indicates that the respondent is unaware of the number of side quests in *The Witcher 3: Wild Hunt*.

A.3 Details of the Synthetic Training Data

Considering the practical application value, we did not synthesize data from the entire corpus. Instead, we randomly sampled a subset of documents for data synthesis. The specific number of sampled documents is shown in the Table 7 under # Selected Docs for Synthesizing Data. In the table, # Original Docs refers to the number of documents obtained after cleaning the raw data. # Selected Docs as Corpus indicates the number of documents used as the retrieval corpus, and # Selected Docs for Constructing Bench refers to the documents used for building the evaluation QA set.

A.4 Detailed Training Parameters

Training settings are as follows: batch size of 32, 3 epochs, learning rate of $2e-5$. All document indices are updated every 5 steps. The loss margin is set to 0.3. For LoRA, the rank is set to 128, alpha is set to 32, and dropout is set to 0.1.

A.5 Expert Annotator Qualifications

Games (Zelda, Hearthstone). We recruited three enthusiasts: one knowledgeable in both domains and two domain-focused. For Zelda, two long-time Nintendo players each reported playing at least four mainline titles (average playtime >100 hours). For Hearthstone, two annotators reported average playtime >300 hours.

Medical (Drug, MayoClinic). We recruited two graduate students in medical fields at a top-tier university. While not experts on every subtopic, both possess foundational biomedical knowledge and at least one focused area relevant to the corpora.

Universities & Documentation (Stanford, Berkeley, Cyotek, Notion). These domains require no deep specialization; four student annotators per dataset were recruited based on prior familiarity (e.g., having browsed the relevant websites or used the software).

Quality control. All annotators followed question-generation and verification rubrics; two rounds of review were conducted for scoring rubrics prior to inclusion in the benchmark.

A.6 CSG Definitions

We list key symbols used in CSG. For query q_i , let \mathcal{A}_i denote the reference answer, $\mathcal{S}(q_i)$ the set of supporting sentences mapped via \mathcal{M}_1 (answer-to-clue) and \mathcal{M}_2 (clue-to-document/sentence), and $w_i \equiv |\mathcal{S}(q_i)|$. The predicate $s \equiv_{\text{LLM}} \mathcal{A}_i$ means the LLM, given the per-question rubric, confirms that sentence s satisfies the criterion linked to \mathcal{A}_i .

A.7 Preliminary Computational Cost Analysis

Synthesis scope. We intentionally synthesize from a subset of documents per corpus (see Table 7), which reduces data-generation overhead while remaining effective empirically.

API/LMM costs. Across all syntheses and filtering steps, the total API expenditure was approximately 200 USD.

A.8 Full Ablation Results Across All Retrievers

This section reports the extended ablations (“w/o logical transformation” and “w/o completeness transformation”) for all retrievers evaluated in the main experiments, across all domains. Please refer to Table 8 for details.

Table 7: Original document count, documents serving as the corpus (i.e., documents in our actual corpus), documents used for data synthesis, documents used for evaluation construction.

	Games		Mediguides		Universities		Documentation	
	Hearthstone	Zelda	Drug	Mayoclinic	Stanford	Berkeley	Cyotek	Notion
# Original Docs	10665	15695	77348	2542	65	280	808	475
# Selected Docs as Corpus	10665	15695	24733	2542	65	280	808	475
# Selected Docs for Synthesizing Data	870	873	826	862	/	/	/	/
# Selected Docs for Constructing Bench	100	100	100	100	65	100	100	100

Table 8: Extended ablation across all evaluated retrievers (Precision@3 \uparrow). Columns correspond to the 8 domain-specific datasets: Zelda, Hearthstone, Drug, Mayo, Stanford, Berkeley, Cyotek, Notion. “Full (ours)” uses both logical and completeness transformations; the two ablations remove one transformation at a time.

Method	Zelda	Hearthstone	Drug	Mayo	Stanford	Berkeley	Cyotek	Notion
gte-base-multilingual								
w/o Logical Transformation	40.03	29.47	49.53	30.42	67.48	50.38	66.23	43.56
w/o Completeness Transformation	38.86	27.34	45.48	28.34	67.34	48.31	65.23	42.34
Full (ours)	41.26	33.67	51.96	30.56	71.48	53.35	68.33	46.62
snowflake-arctic-m-long								
w/o Logical Transformation	33.96	32.13	42.43	24.45	72.48	51.23	63.96	45.32
w/o Completeness Transformation	34.63	34.48	42.13	23.45	72.36	52.13	64.57	43.76
Full (ours)	36.43	36.50	44.58	26.60	75.51	56.22	68.57	48.57
rubert-tiny-turbo								
w/o Logical Transformation	9.10	7.23	9.32	8.12	17.31	11.07	20.13	19.45
w/o Completeness Transformation	9.29	8.23	9.47	8.21	16.13	11.18	17.23	18.54
Full (ours)	9.17	8.67	10.64	8.57	19.26	13.18	22.53	21.65