LongTailQA: Benchmarking LLMs and RAG Models on Disambiguated Long-Tail Entities

William Xion[1], **Uwe Hadler**[1], **Tim Cofala**[1], **Maximilian Idahl**[1], **Soumyadeep Roy**[2], **Wolfgang Nejdl**[1]
[1]L3S Research Center, Hannover, Germany
[2]Department of Medicine (Biomedical Informatics), Stanford University, Stanford, CA, USA
[1]{william.xion, uwe.hadler, tim.cofala, idahl, nejdl}@l3s.de
[2]soroy@stanford.edu

Abstract

Large Language Models (LLMs) struggle with memorizing long-tail facts. Retrieval-Augmented Generation (RAG) models show better performance on long-tail Question Answering (QA) by offloading memory to external knowledge sources. We demonstrate that popular QA benchmarks such as PopQA, WITQA, and EntityQA contain significant entity ambiguity, with 8-30% of long-tail questions referencing entities with non-unique names. This ambiguity confounds evaluation, obscuring true model capabilities. To perform robust benchmarking, we disambiguate these questions with the Wikipedia knowledge graph to develop LongTailQA, an improved QA benchmark that mitigates entity ambiguity in long-tail entity questions. We evaluate various recent LLMs and RAG models, such as Self-RAG and InstructRAG, investigating retriever quality and retrieval depth impacts on QA performance. We observe that: (i) disambiguation improves model accuracy up to 24.7%, (ii) RAG models benefit significantly more than vanilla LLMs, (iii) simply increasing retrieval depth does not improve RAG performance, and (iv) RAG models achieve high accuracy with perfect information, highlighting the need to filter noisy documents during retrieval. The LongTailQA benchmark facilitates robust evaluation of long-tail knowledge recall and RAG system effectiveness. We make the codebase and datasets publicly available at https://github.com/williamx854/LongTailQA-Benchmark.

Keywords: Long-Tail Question Answering, Retrieval-Augmented Generation, Entity Disambiguation

1. Introduction

LLMs have demonstrated impressive capabilities across various tasks, yet they continue to struggle with memorizing and accurately answering questions about long-tail entities and facts (Kandpal et al., 2023; Mallen et al., 2023). RAG has emerged as a primary solution, enhancing LLMs by incorporating external knowledge retrieved from vast corpora, thereby improving question answering capabilities about long-tail entities (Asai et al., 2024; Zhang et al., 2024). Given that the majority of real-world knowledge resides in the long tail, reliably evaluating the performance of both vanilla LLMs and RAG systems on these less popular entities is crucial for understanding their true capabilities and limitations.

However, current evaluation practices are often hampered by inherent flaws within existing long-tail QA benchmarks. Our initial analysis revealed a significant challenge: popular datasets frequently used for this purpose, such as PopQA (Mallen et al., 2023), WITQA (Maekawa et al., 2024), and EntityQA (Sciavolino et al., 2021), contain substantial entity ambiguity. We found that 8-30% of questions in their long-tail subsets reference subject entities with non-unique names (e.g., multiple distinct individuals or works sharing the same name). As we demonstrate in our analysis (Section 4.2.1), this ambiguity significantly confounds evaluation results, potentially underestimating model performance and obscuring the true effectiveness of different approaches. Analyzing performance across different model architectures, we observe substantially lower accuracy on the ambiguous subsets compared to their disambiguated versions, with the performance gap reaching up to 30.0% accuracy for Qwen2.5-7B-Instruct (36.7% versus 66.7% accuracy) on PopQA and 25.0% accuracy for Llama2-13b-Instruct (34.9% versus 59.9% accuracy) for EntityQA. Therefore, as our **first research contribution**, we identify and quantify the impact of this widespread ambiguity problem in the existing entity QA datasets for long-tail entities.

To address this evaluation challenge and enable fairer, more reliable assessments, our **second research contribution** is the construction of the LongTailQA benchmark. It consists of 6537 disambiguated questions from 55 unique relation types and multiple question generation styles (template-based and model-assisted). LongTailQA is constructed by systematically identifying and disambiguating the ambiguous questions within the long-tail subsets (entities with low average monthly Wikipedia page views) of PopQA, EntityQA, and WITQA. Our proposed disambiguation process leverages Wikidata's knowledge graph structure and entity descriptions to resolve ambiguity while preserving the question's investigative nature; further details are provided in Section 3.2.1,. Long-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 5182–5191
11–16 May 2026.

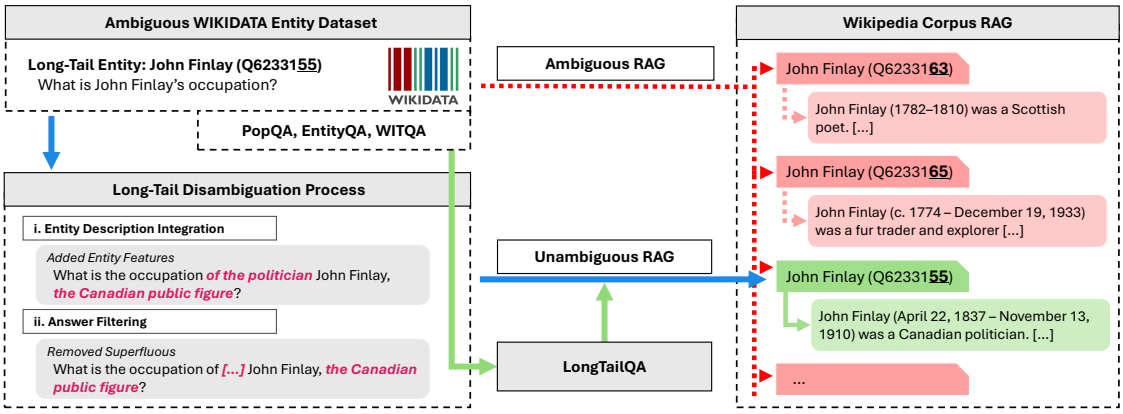

Figure 1: Overview of the proposed disambiguation process constructing the LongTailQA dataset with disambiguated long-tail Wikidata entities. Wikidata entity datasets run through our disambiguation integration process, appending entity descriptions and filtering answer-revealing information

TailQA facilitates fine-grained evaluation of two distinct capabilities in QA systems. For vanilla LLMs without retrieval, it measures whether parametric memory can recall facts about long-tail entities. For RAG systems, it isolates the model's ability to synthesize correct information from retrieved documents, separate from disambiguation challenges. Therefore, LongTailQA will serve as an important resource for the community.

As our **final research contribution**, we evaluate vanilla LLMs such as Llama 2, Llama 3.1, Ministral, Qwen 2.5, GPT-4o and, state-of-the-art RAG models such as Self-RAG (Asai et al., 2024), and InstructRAG (Wei et al., 2025)) on the LongTailQA benchmark. We observe that advanced RAG architectures like InstructRAG-FT achieve the highest performance (up to 90.2% accuracy), substantially outperforming vanilla LLMs (which struggle to exceed 30% accuracy on long-tail knowledge). We also observe that RAG systems generally achieve only 80-85% of the theoretically possible accuracy given the retrieval recall, highlighting opportunities for enhancing how models utilize retrieved information.

Furthermore, our study analyzes key factors influencing RAG performance on the LongTailQA benchmark, including the impact of retrieval depth (varying from 5 to 20) and the models' ability to utilize perfectly relevant information when provided with gold standard documents. We found that simply increasing the number of retrieved documents does not guarantee better performance, with newer models demonstrating greater robustness to potentially noisy contexts. Our gold document experiments establish a practical upper bound of 90.9% accuracy, revealing significant room for improvement in retrieval components. In essence, we conduct extensive experimentation on the LongTailQA benchmark with special focus on the retriever aspect of RAG models.

2. Related Work

Entity-based Question Answering Datasets. Question-answering tasks are fundamental performance evaluation sets. They present knowledge tuples retrieved from Wikidata (PopQA) (Mallen et al., 2023), QA websites (Zhihu-KOL) (wangrui6), focus on long-form ambiguous factoid questions (ALCE-ASQA) (Gao et al., 2023), fact-checking (PubHealth) (Zhang et al., 2023), or biography generation (Min et al., 2023). The latter incorporates the LLM's generation capability, shifting the focus away from evaluating the retrieval performance. Therefore, we focus on the short-form entity datasets PopQA (Mallen et al., 2023), WITQA (Maekawa et al., 2024), and Entity Questions (Sciavolino et al., 2021) due to a simplified evaluation and performance measure and leave long-form generation tasks for future work.

Ambiguity in Language Resources and Retrieval-Augmented Systems. Several recent work examines ambiguity and disambiguation from diverse perspectives. 3AM (Ma et al., 2024) deliberately introduces visual/lexical ambiguity in multimodal machine translation so models must use images, while Wei and King (Wei and King, 2024) study lexical sense shifts over short temporal windows and Singhal et al. (Singhal et al., 2024) generate clarification questions to mitigate vagueness/incompleteness in legal contracts. In contrast, our contribution is a dataset-level resource for open-domain QA that targets a different ambiguity type: referential entity ambiguity where the same surface form can refer to multiple Wikidata entities.

3. Disambiguation and LongTailQA Benchmark Construction

In this section, we first examine the datasets used in our study—PopQA, EntityQA, and WITQA—focusing on their structure, content, and the criteria used to define their long-tail subsets, which collectively form the basis of our LongTailQA benchmark (3.1). We then describe our method to identify and quantify entity ambiguity, followed by the disambiguation pipeline used to construct the final LongTailQA resource (3.2). Finally, we present summary statistics and characteristics of the resulting benchmark (3.2.2).

3.1. Long-tail Subset of Entity-based QA Datasets

Here, we describe the three entity-based QA datasets and the methodology to obtain the long-tail subset from them.

3.1.1. PopQA

PopQA (Mallen et al., 2023) is an entity-centric open-domain QA dataset designed to evaluate factual memorization capabilities of language models across different entity popularity levels. It contains 14,000 entity-related questions constructed using relationship-specific templates applied to Wikidata knowledge triples of the form ⟨*subject entity, relationship, object entity*⟩. The dataset covers 16 relationship types including occupation, place of birth, and author. For example, ⟨*Kathy Saltzman, occupation, politician*⟩ becomes "What is Kathy Saltzman's occupation?"

Long-tail Subset Construction. Each question in the dataset includes metadata for the subject entity, such as its Wikidata ID and monthly Wikipedia page views. We use the 1,399 questions where subject entities have fewer than 100 monthly Wikipedia page views following (Asai et al., 2024).

3.1.2. EntityQA

EntityQA (Sciavolino et al., 2021) was designed to test dense retriever generalization on entity-centric queries. It contains 22,075 template-based questions across 24 relations from Wikidata. Unlike PopQA's person-centric focus, EntityQA includes more diverse entity types with questions like "Where is the headquarter of [E]?" and "Which company is [E] produced by?"

Long-tail Subset Construction. Entity linking was performed to obtain Wikidata IDs, since EntityQA does not include popularity metadata. Each subject's Wikipedia page was then filtered by its 2023 average monthly views. Analyzing the original PopQA long-tail subset under the same metric showed that the <100 threshold used there corresponds to <30 views in 2023. To ensure comparable popularity distributions across benchmarks, we therefore adopt <30 average monthly views as the long-tail threshold for EntityQA, yielding 2,882 questions.

3.1.3. WITQA

WITQA (Maekawa et al., 2024) employs 32 relation types with GPT-3.5-assisted question generation rather than templates. Questions were generated from knowledge triples and supporting Wikipedia passages, then evaluated against criteria (Answerable, HasSubject, NoObject) with up to three regeneration iterations for quality improvement.

Long-tail Subset Construction. We use the 2256 questions where subject entities with <100 monthly page views (provided by the authors). We confirmed this subset's popularity distribution aligns with our <30 threshold used for PopQA and EntityQA. WITQA adds significant variety through its 32 relation types and model-assisted generation.

3.2. Construction of LongTailQA Benchmark

We present a comprehensive analysis of question ambiguity in long-tail entity QA and our steps to mitigate its impact from entity-based QA datasets to form the LongTailQA benchmark. We refer to the long-tail subsets of the previously mentioned datasets simply as PopQA, EntityQA, and WITQA throughout the remainder of this paper.

Entity ambiguity creates a fundamental evaluation problem: questions like "What is John Finlay's occupation?" have ground truth referring to one specific entity (e.g., the Canadian politician, Q6233155), but the question text provides no indication of which John Finlay is intended when multiple entities share this name. Even entity-aware RAG systems (e.g., GraphRAG (Han et al., 2025)) cannot be properly assessed on such benchmarks, as correct answers about alternate entities are systematically marked wrong.

Our initial error analysis with 100 randomly sampled data points that are incorrectly answered by the Self-RAG 13B model on the PopQA dataset revealed that 35% of these errors were due to 'entity ambiguity', a subtype of question ambiguity, where the model provided an answer that was factually correct but pertained to a different entity sharing the same name as the question's subject entity. We believe this ambiguity poses a significant problem in accurately evaluating model performance in long-tail QA: an objectively correct answer (albeit about a different entity) will be labeled as wrong primarily due to insufficient information. We argue that question ambiguity in the benchmark dataset

unfairly penalizes LLM performance. Since these errors arise not from the model failing to produce a correct answer but from ambiguity in the evaluation, the actual capability of the LLM is misrepresented. This finding motivates our rigorous examination of identifying and mitigating entity ambiguity.

3.2.1. Detection and Disambiguation of Entity Ambiguity

Given that the datasets are constructed using the Wikidata knowledge graph, we can leverage its structure to characterize question ambiguity formally. A naive approach is to consider a question ambiguous if its subject entity's surface name matches multiple distinct entities in the knowledge base. However, this surface-level definition proves insufficient as it fails to account for the semantic constraints imposed by the question's predicate. Consider the query, "Who was the author of The Valley?" The subject entity "The Valley" shares its surface name with 36 other distinct entities in Wikidata. Yet despite this apparent lexical ambiguity, only two of these entities contain an "author" property in their knowledge graph structure. While sharing the same name, the remaining entities are fundamentally incompatible with the question's predicate and thus irrelevant to its interpretation.

Definition. Based on predicate compatibility, we formally define *entity ambiguity* to occur when there exists at least one other entity in the knowledge base that (1) shares the same surface name as the subject entity and (2) contains the predicate specified in the question (e.g., the "author" predicate).

Empirical Analysis. Using this definition, we query the Wikidata knowledge graph and find that 463 out of 1,399 questions (33.1%) in PopQA are entity-ambiguous. For EntityQA and WITQA, we found 441 out of 2882 and 183 out of 2256 respectively (15.3% and 8.1% respectively) are ambiguous. This substantial proportion suggests that ambiguity is a significant challenge for LLMs answering questions about long-tail entities. For PopQA, we perform some manual inspection to understand which relation types have more ambiguous queries. We find that questions with "director," "producer," and "genre" predicates exhibit the highest ambiguity rates (49.5%, 40.5%, and 47.6%, respectively). Based on manual inspection of these cases, we believe this high level of ambiguity happens because certain names in these categories, like "The Valley," are too generic and can easily refer to multiple entities.

Disambiguation of Datasets. To create clean versions of the datasets, we apply a straightforward disambiguation approach using Wikidata entity descriptions. We refer to the cleaned versions as *DisambPopQA*, *DisambEntityQA*, and *DisambWITQA*. We use GPT-4o with few-shot prompting to append relevant Wikidata entity descriptions to questions identified as ambiguous. For example, "World of Wonder" (Q8036644) has the description "anthology of sci-fi and fantasy short stories", which helps distinguish it from other entities with the same name. For certain Wikidata entities, we observe that the subject entity's Wikidata description contains the answer to the query; thus, for the initial prompting, we also specifically ask GPT4o to not incorporate any information that might contain the answer. After this, we run one more round of filtering with GPT4o to ensure the augmented questions do not contain the answer. The final benchmark preserves the original unambiguous questions and incorporates the disambiguated versions of questions identified as ambiguous.

Dataset	Relations	Form	Count
DisambEntityQA	23	Template	2882
DisambWITQA	32	Model Assist.	2256
DisambPopQA	15	Template	1399
LongTailQA	55	Mixed	6537

Table 1: Dataset Statistics of LongTailQA

3.2.2. LongTailQA Benchmark Statistics

The final LongTailQA benchmark is constructed by identifying, disambiguating, and combining the long-tail subsets of the three entity question-answering datasets: PopQA, EntityQA, and WITQA. The composition of the resulting benchmark is detailed in Table 1. LongTailQA consists of 6,537 questions, offering a substantial testbed for evaluating long-tail QA performance. The benchmark incorporates questions derived from 55 unique Wikidata relation types, ensuring diversity beyond just common relations, largely due to the inclusion of WITQA, which itself covers 32 distinct types. Furthermore, the benchmark benefits from varied question generation methodologies; while the PopQA and EntityQA components rely on template-based questions, the WITQA component utilizes questions generated and refined via a model-assisted approach, offering greater linguistic variety.

4. Benchmarking Study with LLMs and RAG Models using LongTailQA

This section details our experimental setup and presents a comprehensive benchmarking study evaluating various LLMs and RAG models on the LongTailQA dataset. The study quantifies the impact of query disambiguation, analyzes RAG model performance with increasing retrieval depth, and

assesses performance when provided with gold documents.

4.1. Evaluation Setup

The primary metric is question answering accuracy, measured using case-insensitive substring matching against the ground-truth answer list(s). For RAG systems, we also report retrieval Recall@5. This measures whether any of the top 5 retrieved documents contain the ground-truth answer, determined via case-insensitive substring matching. Knowing the Recall@5 alongside accuracy lets us assess how effectively RAG systems utilize potentially relevant information provided by the retriever. The retriever used for all RAG evaluations (unless otherwise specified for a particular advanced RAG system) is Contriever-MS MARCO (Izacard et al., 2022), searching over the December 2020 pre-processed Wikipedia corpus from Izacard et al. (Izacard et al., 2023).

4.1.1. Baseline Models

Our baseline LLMs include instruction-tuned variants from different families and generations: Llama 2 (7B-Instruct, 13B-Instruct) (Touvron et al., 2023), Llama 3.1 (8B-Instruct) (Dubey et al., 2024), Ministral (8B-Instruct) (AI, 2024) and Qwen 2.5 (7B-Instruct) (Yang et al., 2024). Among the different recent RAG solutions, we selected two state-of-the-art approaches: Self-RAG (Asai et al., 2024) and InstructRAG (Wei et al., 2025). InstructRAG self-reflects on retrieved documents using rationales, while Self-RAG (Asai et al., 2024) reflects on retrieved documents and additionally reflects on the necessity of retrieval. Our evaluated RAG systems include: (i) Self-RAG (7B, 13B, based on Llama 2) (Asai et al., 2024), and (ii) InstructRAG (based on Llama 3)(Wei et al., 2025).

InstructRAG. Standard RAG faces a known challenge: retrieved documents often contain irrelevant or contradictory information due to imperfect retrieval or noisy source data (Shi et al., 2023). To address this, InstructRAG (Wei et al., 2025) proposes a framework to explicitly denoise retrieved context using self-synthesized rationales. InstructRAG employs a two-step methodology: first, an instruction-tuned LLM generates a rationale explaining how the answer derives from potentially noisy documents; second, these rationales teach a target LLM the denoising skill. This occurs either via in-context learning (InstructRAG-ICL) or through fine-tuning (InstructRAG-FT), both aiming to produce more trustworthy answers than standard RAG.

Self-RAG. Self-RAG (Asai et al., 2024) is an adaptive RAG system that determines when to retrieve documents and reflects on their relevance. Unlike traditional RAG, it processes retrieved documents in parallel, generating answers for each document and using critique tokens to score and select the best response. The system employs three critique tokens: relevance (binary), support (3-point scale), and utility (5-point scale) to evaluate document quality and answer generation.

Model	Original Acc [%]	Disamb. Acc [%]
Non-Retrieval LLMs		
Llama2 7B-Instruct	21.6	26.6
Llama2 13B-Instruct	15.7	23.2
Llama3.1 8B-Instruct	27.3	30.1
Qwen2.5 7B-Instruct	25.9	32.7
Ministral 8B-Instruct	27.0	29.7
GPT-4o	**43.5**	**54.9**
RAG @ k=5		
Llama2 7B-Instruct	38.3	63.0
Llama2 13B-Instruct	39.9	63.4
Self-RAG 7B	50.1	68.1
Self-RAG 13B	47.2	68.2
Llama3.1 8B-Instruct	**73.1**	79.1
Qwen2.5 7B-Instruct	53.6	71.6
Ministral 8B-Instruct	69.9	77.6
InstructRAG ICL	62.5	77.8
InstructRAG FT	65.2	**80.2**

Table 2: Impact of disambiguation on accuracy (averaged across datasets). Resolving entity ambiguity yields substantial accuracy improvements across nearly all models, confirming that ambiguity significantly confounds evaluation in the original benchmarks. Best results are highlighted in bold

4.1.2. Implementation Details

For the inference parameters, we aim for consistency where possible while respecting model-specific recommendations. For Self-RAG, we used the default scoring weights, temperature 0.0, and top-p 1.0 as specified by Asai et al. (Asai et al., 2024). For all other baseline LLMs, we used a temperature of 0.8 and top-p 0.95. Unless explicitly stated otherwise, these parameters were used across all experiments.

4.2. Experimental Results

We now examine how existing LLMs and RAG models perform on long-tail entity questions.

4.2.1. Performance Improvement due to Query Disambiguation

Before presenting the main results on the fully disambiguated LongTailQA benchmark, we first quantify the direct impact of our disambiguation process. As discussed in 3.2.1, a portion of the original long-tail questions in PopQA, EntityQA, and WITQA suffered from entity ambiguity. To isolate the effect

of resolving this ambiguity, we compare the performance of a representative set of models on the subset of questions originally identified as ambiguous versus their performance on the corresponding disambiguated versions of those same questions. The accuracy averaged across the three source datasets for these subsets is presented in Table 2.

The results clearly demonstrate that entity ambiguity hurts the performance of both vanilla LLMs and advanced RAG systems. As shown in Table 2, all evaluated models exhibit substantial accuracy improvements after the ambiguous queries are disambiguated. For instance, the closed-source GPT-4o improved by 11.4% percentage points on average. Among the open-source instruction-tuned models, Qwen2.5-7B-Instruct gained 6.8% average points, while the older Llama2-13B-Instruct saw a 7.5% average point increase.

The effect is equally pronounced for RAG systems operating at k=5 retrieval depth: Self-RAG 13b improved by 15% points, Qwen2.5-7B-Instruct (with retrieval) by 17.9% average points, and InstructRag-ICL by 15.3% average points. Notably, the average performance gain for retrieval-augmented (RAG) models (13.2% points) is substantially higher than for non-retrieval LLMs (4.4% points), indicating that disambiguated queries particularly benefit retrieval-based approaches. These findings validate our disambiguation methodology and highlight the necessity of using cleaned datasets for reliable evaluation. The substantial performance gap confirms that ambiguity acts as a major confounder, leading to inaccurate conclusions about model capabilities on long-tail QA. Therefore, the subsequent analyses in this paper focus on the fully disambiguated LongTailQA benchmark. Disambiguation not only improved downstream accuracy but also significantly boosted retrieval effectiveness. Contriever-MS MARCO yields 69.5% average Recall@5 on ambiguous subsets, rising to 82.4% after disambiguation.

4.2.2. Benchmarking Study of LLMs and RAGs on LongTailQA

We now present the main performance results on the fully disambiguated LongTailQA benchmark. Table 3 summarizes the accuracy of various Large Language Models (LLMs) operating without retrieval, alongside the accuracy and average retrieval recall (Recall@5) for Retrieval-Augmented Generation (RAG) systems retrieving the top 5 documents using Contriever-MS MARCO(Izacard et al., 2022).

Table 3 reveals several key insights into model performance on the disambiguated LongTailQA benchmark. First, even the powerful closed-source GPT-4o achieves only 55.5% average accuracy, while open-source models like Llama 2 perform below 26% on average. This underscores the limitations of relying solely on parametric memory for accessing less popular facts. Second, Retrieval-Augmented Generation (RAG) provides a substantial performance boost. Both non-instruction tuned and instruction tuned models show performance increase, with instruction tuned models showing a much larger increase (e.g., Qwen2.5-7B-Instruct jumps from 31.5% to 79.2% in terms of average accuracy, a 47.7 percentage point increase), demonstrating the critical role of external knowledge retrieval for long-tail QA. Third, newer generation open-source models generally outperform older ones. Comparing Llama 3.1 and Qwen 2.5 against Llama 2 shows clear advantages in both non-retrieval and RAG settings, highlighting progress in base model capabilities. Qwen2.5-7B-Instruct, both with and without RAG, emerges as a particularly strong performer among the evaluated open-source models of its size class. Fourth, advanced RAG architectures show further benefits. InstructRAG-FT achieves the highest average accuracy (85.2%), surpassing other systems, including Self-RAG (average accuracy of 73.2%) and standard RAG configurations like Qwen2.5+RAG (accuracy of 79.2%). This suggests that specialized training or architectures for retrieval interaction, like

Model	Accuracy [%]		
	Disamb EntityQA	Disamb WITQA	Disamb PopQA
Non-Retrieval LLMs			
Llama2 7B-Instruct	15.7	34.8	24.9
Llama2 13B-Instruct	11.4	31.6	21.8
Llama3.1 8B-Instruct	23.5	38.6	28.7
Qwen2.5 7B-Instruct	24.6	40.5	29.4
Ministral 8B-Instruct	22.0	39.5	28.9
GPT-4o	**47.0**	**66.9**	**52.7**
RAG @ k=5			
Llama2 7B-Instruct	66.6	77.6	67.3
Llama2 13B-Instruct	65.6	76.5	71.6
Self-RAG 7B	68.4	81.3	70.5
Self-RAG 13B	68.3	80.9	69.6
Llama3.1 8B-Instruct	78.7	85.5	76.0
Qwen2.5 7B-Instruct	76.6	85.4	75.6
Ministral 8B-Instruct	76.4	84.8	75.5
InstructRAG ICL	79.1	88.6	80.3
InstructRAG FT	**83.4**	**90.2**	**82.1**

Table 3: Performance overview of the disambiguated LongTailQA benchmark on various LLMs and RAG systems. RAG models use k=5 retrieval with Contriever-MS MARCO. RAG significantly boosts long-tail QA performance over vanilla LLMs. Newer RAG systems like InstructRAG-FT achieve the highest accuracy, nearly matching retrieval recall, though a notable gap between recall and accuracy persists for many models. Best results are written in bold

those in InstructRAG, are effective.

Across datasets, models generally achieve their highest scores on DisambWITQA, most likely due to the use of GPT3.5 for round-trip refinement to make the question more answerable and natural-sounding. Performance is typically lowest on DisambEntityQA. However, relative model rankings remain largely consistent across the three datasets. Finally, comparing RAG accuracy with the average Recall@5 of 86.9% reveals a performance gap. While the top-performing InstructRAG-FT nearly closes this gap (Avg. Acc 85.2%), other systems like Self-RAG (Avg. Acc 73.2%) show considerable room for improvement in effectively utilizing the retrieved information, even when the correct answer is present in the top 5 documents.

4.2.3. Effect of Retrieval Depth on RAG Performance

While the main results in Section 4.2.2 focus on the standard RAG setting with k=5 retrieved documents, increasing the number of documents (k) can potentially improve performance by increasing the likelihood of retrieving relevant information (recall). However, processing more documents also increases context length and the potential for distraction by irrelevant information. To investigate this trade-off on the LongTailQA benchmark, we analyze the performance of several RAG systems when varying the number of retrieved documents from k=5 to k=20. When the value of 'k' is increased, the retrieval recall (averaged across the three datasets) increases consistently across all the models: 86.9% at k=5, 89.3% at k=10, 90.4% at k=15, and 91.0% at k=20. However, Figure 2 shows that this improved recall does not uniformly result into better accuracy across all the models.

The standard RAG configuration using the newer Qwen2.5-7B-Instruct base model demonstrates robustness; its accuracy continues to improve slightly as k increases, reaching its peak at k=20 (81.3%). This suggests it can effectively leverage the marginally higher recall without being significantly hampered by the additional documents. Similarly, the advanced InstructRAG-FT maintains high accuracy, peaking at k=10 (86.0%) and only slightly decreasing at k=20 (82.6%), indicating strong performance in utilizing relevant information even within a larger context. In contrast, models based on the older Llama 2 architecture exhibit different trends.

The instruction-tuned version, Llama2-13B-Instruct RAG setup, improves slightly up to k=15 (72.9%). We did not evaluate Llama2-13B-Instruct at k=20 due to context window limitations. Self-RAG 13B, although based on Llama 2, process each individual document in parallel so did not have context window problems. Self-RAG shows a consistent decline in accuracy as k increases beyond 5, dropping from 68.3% (k=5) to 66.6% (k=20), despite being able to process the input. This suggests that its mechanism for selecting the best response among parallel generations may struggle with a larger number of candidates or that the base Llama 2 model itself is more sensitive to distraction, outweighing the benefit of higher recall.

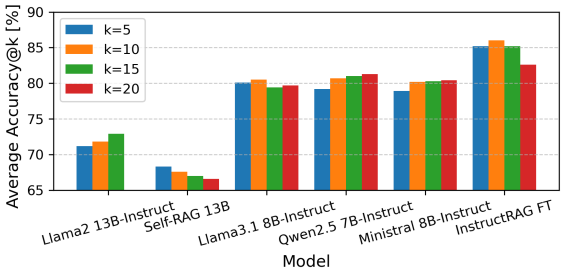

Figure 2: Average Accuracy@k [%] on LongTailQA for selected RAG models as retrieval depth (k) increases. Llama2 was not evaluated at k=20 due to context window limitations. Although increasing retrieval depth consistently improves recall, end-to-end accuracy does not uniformly increase, highlighting varying model robustness to longer contexts and noise

These results show that increasing retrieval depth does not guarantee better RAG performance. While newer models like Qwen2.5 and specialized architectures like InstructRAG appear more robust, older models or specific architectures like Self-RAG can suffer performance degradation, likely due to challenges in processing longer contexts or effectively identifying the correct answer among the retrieved passages, even as retrieval recall improves.

4.2.4. Performance Evaluation in Perfect Retrieval Setting

We conduct experiments where models are provided with "gold" knowledge documents to establish a practical upper bound for RAG performance on the LongTailQA benchmark and assess how well models can utilize perfectly relevant information. We query the Wikidata knowledge graph to obtain the gold knowledge documents for each subject entity in LongTailQA. We then extract the URL of the corresponding Wikipedia article and scrape the articles' raw text from all paragraphs (excluding elements like infoboxes or references). We use the full scraped text as the gold document for each question, treating it as a single passage. We found that 0.73% (n=48) of subject entities' Wikipedia articles have since been deleted. For these cases, we manually add a one-sentence gold document containing the correct answer.

We evaluate model performance under two con-

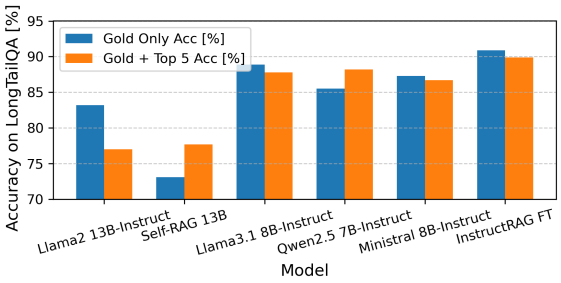

Figure 3: Average accuracy [%] on LongTailQA with gold document context. While models achieve high accuracy with perfect information (Gold Only), performance generally degrades slightly for most models when handling noise (Gold + Top 5), revealing differences in model robustness

ditions involving these gold documents: (i) *Gold Only (k=1):* The model receives only the single gold document as context. This isolates the model's ability to extract the answer given perfect information. (ii) *Gold + Top 5 (k=6):* The model receives the top 5 documents retrieved by Contriever-MS MARCO (as used in Section 4.2.2) plus the gold document, which is inserted as the first document in the context. This tests the model's ability to identify and prioritize the gold information among potentially noisy retrieved documents.

The results in Figure 3 show that providing gold documents significantly boosts performance compared to standard RAG (Table 3), establishing an upper bound. In the "Gold Only" setting, most models achieve over 80% average accuracy, with InstructRAG-FT reaching 90.9%, demonstrating strong capabilities in extracting answers when presented with perfect context. Interestingly, the performance of Self-RAG 13B in the "Gold Only" setting (73.1%) is lower than might be expected. We found the reason for this is due to the model frequently outputting empty strings (7.9% on EntityQA, 7.1% on WITQA, and 26.7% on PopQA). This issue is less pronounced when multiple documents are retrieved (k=6 setting) due to its parallel processing architecture; if one generation path results in an empty string, answers generated from other retrieved documents can still be selected as the final output. Consequently, Self-RAG 13B paradoxically performs better in the k=6 setting (77.7%) than in the k=1 setting (73.1%).

Comparing the "Gold Only" and "Gold + Top 5" settings reveals how models handle distraction. Most models experience a performance drop when five potentially noisy retrieved documents are added, indicating difficulty in consistently prioritizing the gold document. For instance, Llama2-13B-Instruct drops from 83.2% to 77.0%, while even top performers like InstructRAG-FT see a slight decrease (90.9% to 89.9%). Newer models like Qwen2.5, Llama3.1, and Ministral show relatively smaller drops compared to Llama 2 variants, suggesting better robustness. These results indicate that while current models are capable of high accuracy with perfect information, challenges remain in robustly identifying and utilizing the most relevant document when presented with a mix of relevant and potentially irrelevant retrieved context. The gap between the "Gold + Top 5" results and the standard RAG with retrieval at 5 results further highlights the significant room for improvement in the retrieval component.

5. Conclusion

This paper addresses a significant evaluation issue in long-tail entity question answering and introduces the LongTailQA benchmark. We identify that 8-30% of questions in popular QA datasets (PopQA, EntityQA, WITQA) suffer from entity ambiguity, where multiple entities share the same name, leading to unfair performance penalties. To create a cleaner evaluation resource, we disambiguate the ambiguous questions using Wikidata entity descriptions. The resulting LongTailQA benchmark comprises 6,537 disambiguated questions across 55 relation types, providing a more reliable testbed for long-tail knowledge evaluation. Our comprehensive benchmarking study demonstrates that disambiguation substantially improves model performance, with RAG systems benefiting more than vanilla LLMs. The benchmark reveals key insights about retrieval depth effects and model robustness, establishing LongTailQA as a valuable resource for evaluating and improving long-tail knowledge handling in both LLMs and RAG systems.

Future Work Building on this LongTailQA, future work can extend the benchmark to multi-hop and compositional reasoning over long-tail entities. Such questions require connecting multiple rare facts across documents, offering a deeper test of retrieval and reasoning capabilities. Future studies can evaluate these tasks using advanced multi-hop RAG architectures and reasoning-oriented LLMs to better understand how well current systems integrate rare knowledge across reasoning steps. It remains unclear how recent advances in retriever methods—such as dense hybrid retrieval or query rewriting—interact with long-tail entities, and future work can systematically examine their effectiveness under long-tail conditions. Another direction is to explore alternative disambiguation strategies for cases where questions are not grounded in a structured knowledge base like Wikidata, including open-ended or user-generated queries.

Data and Code Availability

The LongTailQA benchmark dataset, along with the evaluation codebase and all prompts used for the LLM and RAG experiments, are publicly available at https://github.com/williamx854/LongTailQA-Benchmark. The dataset is also available on Hugging Face at https://huggingface.co/datasets/willx7890/LongTailQA.

Limitations

Despite our analysis's valuable insights into ambiguity's impact on retrieval and language model performance, several limitations deserve attention. Our study's focus on entity-centric, template-generated questions represents only a subset of the broader long-tail question-answering domain, encompassing more complex semantic structures and multiple ambiguity layers. The heavy reliance of our query enrichment methodology on structured knowledge bases, particularly Wikidata, presents another limitation. We do not compare different disambiguation strategies (e.g., GraphRAG, entity linking methods) as LongTailQA is designed as an evaluation resource rather than a disambiguation method. Instead, our experiments establish baseline performance across model architectures and analyze factors affecting RAG effectiveness such as retrieval depth, context noise, gold document utilization.

Acknowledgments

This work was partially funded by the Bundesministerium für Wirtschaft und Energie (BMWE), Germany, in the context of the 8ra Initiative ("Soofi", 13IPC040E).

Bibliographical References

Mistral AI. 2024. Un ministral, des ministraux.

Akari Asai, Zeqiu Wu, Yizhong Wang, et al. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6465–6488. Association for Computational Linguistics.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2025. Retrieval-augmented generation with graphs (graphrag). *CoRR*, abs/2501.00309.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, et al. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, et al. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Nikhil Kandpal, Haikang Deng, Adam Roberts, et al. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, et al. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. 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*, pages 6138–6148. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, et al. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Anmol Singhal, Chirag Jain, Preethu Rose Anish, Arkajyoti Chakraborty, and Smita Ghaisas.

2024. Generating clarification questions for disambiguating contracts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 7611–7622. ELRA and ICCL.

Hugo Touvron, Louis Martin, Kevin Stone, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

wangrui6. GitHub - wangrui6/Zhihu-KOL — github.com. https://github.com/wangrui6/Zhihu-KOL. [Accessed 02-12-2024].

Yuchen Wei and Milton King. 2024. Sense of the day: Short timeframe temporal-aware word sense disambiguation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 14676–14686. ELRA and ICCL.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales.

An Yang, Baosong Yang, Beichen Zhang, et al. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, et al. 2023. Interpretable unified language checking. *CoRR*, abs/2304.03728.

Zihan Zhang, Meng Fang, and Ling Chen. 2024. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6963–6975. Association for Computational Linguistics.

Language Resource References

Xinyu Ma, Xuebo Liu, Derek F. Wong, Jun Rao, Bei Li, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min Zhang. 2024. 3am: An ambiguity-aware multi-modal machine translation dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 1–13. ELRA and ICCL.

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*, pages 5506–5521. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, et al. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. 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*, pages 6138–6148. Association for Computational Linguistics.