DEEPAMBIGQA: A BENCHMARK FOR AMBIGUITY-AWARE DEEP RESEARCH QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep-research question answering requires long multi-hop reasoning and precise information retrieval from large contexts. Recent large language models (LLMs) demonstrate strong performance, yet still face challenges in answer completeness, especially when reasoning requires aggregating evidence across large candidate pools. For example, the question *"Which actors have collaborated with both Hans Zimmer and Nolan?"* is difficult because *"Nolan"* may refer to either *Christopher* or *Jonathan Nolan*, necessitating careful disambiguation and exhaustive cross-checking of all possible actors. In this paper, we introduce DEEPAMBIGQA, a benchmark of automatically generated deep-research questions deliberately designed to induce ambiguity while demanding multi-hop reasoning over extensive candidate sets. Our generation pipeline produces challenging questions along two axes: ❶ referential ambiguity, where shared names correspond to distinct entities and yield divergent correct answers, and ❷ aggregation ambiguity, where solutions require collecting and integrating evidence across large answer spaces. Our evaluation shows that frontier LLMs underperform on DEEPAMBIGQA, with errors primarily driven by incomplete retrieval and poor disambiguation. We further introduce disambiguation and context-reduction modules, which improve performance but still find DEEPAMBIGQA difficult to solve. By isolating ambiguity as a key limiting factor, we provide a focused resource for advancing deep-research QA.

1 INTRODUCTION

Deep-research question answering (QA) aims to answer complex, information-seeking queries that require reasoning over multiple sources of evidence. Unlike factoid QA, which often resolves with a single lookup, deep-research QA questions are inherently *multi-hop*, requiring systems to decompose queries into intermediate steps, retrieve evidence from diverse and often large contexts, and integrate it into a coherent answer. This ability is critical for building systems that can support real-world investigative tasks, from scientific literature review to business intelligence (Lee et al., 2024). Large language models (LLMs) have recently demonstrated impressive progress on this front, achieving strong results on benchmarks such as HotpotQA (Yang et al., 2018), QASC (Khot et al., 2020), and MuSiQue (Trivedi et al., 2022). However, the effectiveness of these systems depends not only on compositional reasoning but also on their capacity to retrieve and assemble complete supporting evidence, a dimension underexplored in current datasets.

Consider the query: *"Among the movies collaborated by Hans Zimmer and Nolan, list all actors that appeared multiple times."* This example illustrates two distinct forms of ambiguity that challenge current systems. First, **referential ambiguity** arises because "*Nolan*" can denote either *Christopher Nolan* or *Jonathan Nolan*, each leading to different sets of candidate movies. Second, **aggregation ambiguity** emerges from the intermediate reasoning step: enumerating all movies co-created with *Hans Zimmer* and *Nolan* yields a large candidate pool, from which the system must accurately retrieve and aggregate recurring actors. Together, these two ambiguity types highlight systematic weaknesses in retrieval and disambiguation that are not exposed by existing benchmarks. Figure 1 provides an overview comparing such deep-ambiguous questions with factoid questions and deep research questions. To resolve these questions, the LLM must track multiple reasoning path branches depending on the entity resolution of the ambiguous name, and keep precise information collection from large contexts during the multi-hop reasoning process.

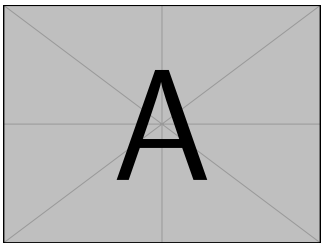

Figure 1: Comparison between DEEPAMBIGQA question (right), factoid QA question, and deep-research question (left). DEEPAMBIGQA question requires correct disambiguation of the key entity for reasoning path branching and precise information keeping during the multi-hop reasoning process.

Constructing such queries and their ground-truth answers is difficult, as it requires extensive manual effort to enumerate large candidate spaces and resolve ambiguous references. Prior datasets involving complex multi-hop reasoning, such as HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022), rely on heavy annotation pipelines with significant human supervision. In contrast, we design a bottom-up **automatic pipeline** for generating and verifying ambiguity-driven multi-hop questions at scale. Our pipeline leverages the structured knowledge of Wikidata (**?**), first identifying seed relations and entities, then composing multi-hop queries that intentionally include referential overlap or large aggregation steps. Candidate questions are automatically generated using LLMs, followed by verification and filtering to ensure answerability. This process yields DEEPAMBIGQA, a benchmark of 1043 questions spanning diverse domains and systematically covering both referential and aggregation ambiguity.

We evaluate state-of-the-art LLMs on DEEPAMBIGQA and find that they struggle to resolve these questions, even for strong reasoning LLMs. Performance degrades primarily due to incomplete evidence retrieval and failures in disambiguation. To probe possible directions forward, we introduce two additional modules: an entity disambiguation mechanism and a retrieval aggregation filter, which yield measurable improvements but still fall short of fully solving the task. These findings underscore the challenge of resolving ambiguities in deep-research QA system.

In summary, our contributions are:

- We design an automatic generation and verification pipeline, combining Wikidata and LLMs, to construct DEEPAMBIGQA, a benchmark of ambiguity-driven multi-hop research questions.
- We evaluate state-of-the-art LLMs and show that they underperform on DEEPAMBIGQA, revealing systematic weaknesses in ambiguity resolution.
- We further propose two modules to address the limitations of existing LLM-powered deep-reserch QA with performance improvement but still highlight the remaining gap.

2 RELATED WORK

2.1 DEEP-RESEARCH QA DATASETS

Open-domain and deep-research QA typically follow a retrieve-then-read paradigm (Chen et al., 2017), with dense retrieval (Karpukhin et al., 2020) and retrieval-augmented models (Guu et al., 2020; Lewis et al., 2020) improving evidence access at scale. Generative readers such as Fusion-in-Decoder (FiD) aggregate many passages effectively (Izacard & Grave, 2021), while multi-hop retrieval architectures specifically target compositional queries (Xiong et al., 2021). Beyond these, large retrieval-augmented LMs demonstrate strong knowledge-intensive performance and updatability—e.g., Atlas and RETRO (Izacard et al., 2023; Borgeaud et al., 2022). Recent controllers and training paradigms make retrieval adaptive, reflective, and more faithful (e.g., Self-RAG and HyDE) (Asai et al., 2023; Gao et al., 2023). Decision-time prompting that decomposes questions and interleaves tool use with reasoning further narrows the research gap: Self-Ask with Search, WebGPT, and Toolformer (Press et al., 2022; Nakano et al., 2021; Schick et al., 2023). Benchmarks

continue to probe compositionality and evidence provenance at scale: in addition to HotpotQA, QASC, MuSiQue, WikiHop, ComplexWebQuestions, and StrategyQA (Yang et al., 2018; Khot et al., 2020; Trivedi et al., 2022; Welbl et al., 2018; Talmor & Berant, 2018; Geva et al., 2021), 2WikiMultiHopQA enforces explicit multi-step reasoning paths (Ho et al., 2020), while KILT unifies knowledge-intensive tasks with shared corpora and provenance requirements (Petroni et al., 2021). Despite these advances, most settings assume that relevant evidence can be retrieved unambiguously and in manageable quantities. Our work isolates ambiguity as the primary obstacle in deep-research QA, emphasizing referential ambiguity and aggregation over large candidate sets.

2.2 Ambiguous QA Datasets

Ambiguity arises when queries admit multiple plausible interpretations or when evidence is incomplete. Natural Questions surfaces multiple valid answer variants and unanswerable cases (Kwiatkowski et al., 2019), while SQuAD 2.0 explicitly includes unanswerable questions (Rajpurkar et al., 2018). AmbigQA centers ambiguity by collecting questions with multiple valid readings and corresponding disambiguated answers, with ASQA extending to long-form summaries that reconcile interpretations (Min et al., 2020; Stelmakh et al., 2022). A complementary line targets incomplete evidence and cross-document gaps (IIRC) (Ferguson et al., 2020). Conversational datasets aim to resolve underspecification through clarification and rewriting: ClariQ and Qulac for clarifying questions; CANARD and QReCC for context-dependent question rewriting; and OR-QuAC for open-retrieval conversational QA (Aliannejadi et al., 2020; 2019; Elgohary et al., 2019; Anantha et al., 2021; Qu et al., 2020). Referential ambiguity is often addressed via entity linking (e.g., BLINK; GENRE; ELQ; Bootleg) (Wu et al., 2020; De Cao et al., 2021; Li et al., 2020; Orr et al., 2021). Finally, QED provides structured, span- and entity-level explanations that make ambiguity and reference explicit (Lamm et al., 2021). In contrast, our benchmark constructs inherently ambiguous *multi-hop* research questions and scales the intermediate aggregation step, exposing retrieval failures that persist even with strong reasoners and modern retrieval augmentation.

3 DeepAmbigQAGen: An Automatic QA Synthesis Pipeline

In this section, we present DeepAmbigQAGen, a bottom-up algorithm for automatically generating complex multi-hop reasoning questions that incorporate diverse forms of ambiguity. Section 3.1 outlines the overall question generation pipeline, and Section 3.2 provides additional details, including methods for filtering invalid questions and tracking referentially ambiguous ones.

3.1 DeepAmbigQAGen Algorithm

We present DeepAmbigQAGen, a bottom-up generative pipeline that synthesizes complex, potentially ambiguous, multi-hop reasoning questions over large text corpora. The method operates by composing *atomic units* into progressively richer query states while enforcing type and semantic constraints against a knowledge graph aligned with the corpus (e.g., Wikipedia–Wikidata).

Formally, let $\mathcal{E}$ denote the universe of entities, $\mathcal{R}$ the set of relations, and $\mathcal{P}$ the space of predicates (e.g., numeric or categorical constraints). For each $e \in \mathcal{E}$, let $e_R \subseteq \mathcal{R}$ be the relations incident to e. An atomic unit is a pair (e, r) with $e \in \mathcal{E}$ and $r \in e_R$. This unit induces a selection set

$$\mathrm{Sel}(e, r) \;=\; \{\, e' \in \mathcal{E} : (e, r, e') \in \mathcal{G} \,\},$$

where $\mathcal{G}$ is the corpus-aligned knowledge graph. The query state at depth t is denoted q_t and is updated by applying one operation from the operation set O = {*Select, Filter, Union, Difference, Intersection, Join, GroupBy, OrderBy*}. If an operation requires a predicate (e.g., `Filter`, `OrderBy`), the pipeline samples $p_t \in \mathcal{P}$; if it requires a second atomic unit, it samples (e', r') analogously. A validator $\texttt{IsValid}(\cdot)$ checks type compatibility and semantic coherence of the composition with respect to $\mathcal{G}$. Upon validation, the state is updated by

$$q_{t+1} \;=\; \texttt{Combine}\big(q_t,\, o_t,\, (e', r'),\, p_t\big).$$

After applying DeepAmbigQAGen to obtain a structured SQL-like description of the problem, we prompt LLM to translate these structured language to a natural language question. Table 1 illusrates the example synthesized query, and the operations involved. As shown in the table, resolving these questions involve multiple reasoning steps for information collecting.

Table 1: Example questions synthesized by DEEPAMBIGQAGEN.

Synthesized Query	Atomic Entities	Operations	Final Answers
Which directors have worked with Saoirse Ronan on two or more films?	Saoirse Ronan	(1) Select film by Saoirse Ronan. (2) Group film by director. (3) Filter director by count.	{Joe Wright, Greta Gerwig, Wes Anderson}
List all actors who have starred in both a Martin Scorsese film and a Quentin Tarantino film.	Martin Scorsese, Quentin Tarantino	(1) Select films by Scorsese. (2) Join casts → actors. (3) Select films by Tarantino. (4) Join casts → actors. (5) Set-intersection actors.	{Robert De Niro, Leonardo DiCaprio, Harvey Keitel, Jonah Hill, Brad Pitt, Margot Robbie, Ray Liotta}
Which directors have directed a film that won Best Picture and also a film starring Meryl Streep	Best Picture, Meryl Streep	(1) Select Best Picture films → directors. (2) Select films starring Meryl Streep → directors. (3) Set-intersection directors. (4) Filter director names.	{Steven Spielberg, Clint Eastwood, Mike Nichols, Alan J. Pakula, Rob Marshall, Sydney Pollack, Jonathan Demme}
Which actors have appeared in a film directed by Christopher Nolan, co-starred with Tom Hardy, and later won an Oscar	Christopher Nolan, Tom Hardy, Oscar	(1) Select films directed by Nolan → actors. (2) Aggregate min collab year per actor. (3) Select actors who co-starred with Tom Hardy. (4) Join Oscar wins. (5) Filter win.year > min collab year. (6) Filter actor names.	{Leonardo DiCaprio, Cillian Murphy, Christian Bale, Mark Rylance}
List all directors who worked with Leonardo DiCaprio, released a film in the Drama genre, and had that film nominated for Best Picture	Leonardo DiCaprio, Drama genre, Best Picture	(1) Select directors who have films with DiCaprio. (2) Filter films by genre = Drama. (3) Join with Best Picture nominations. (4) Set-intersection by director. (5) Filter director names.	{Martin Scorsese, Alejandro G. Iñárritu, Quentin Tarantino, Steven Spielberg, Baz Luhrmann, Edward Zwick, Lasse Hallström}

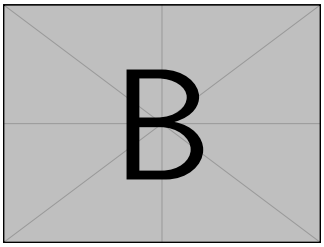

Figure 2: Illustration of the DEEPAMBIGQAGEN algorithm. Beginning with a large collection of entities and relations, a complex query is constructed by iteratively sampling atomic entities and new operations. The resulting query is then converted into natural language by prompting an LLM.

3.2 OTHER DETAILS

Invalid Question Filtering A central challenge in automatic query generation is the prevention of semantically incoherent or unanswerable questions. During each composition step, the pipeline invokes `IsValid`$(\cdot)$ to ensure well-formedness. In this process, we employ a hybrid approach combining heuristic rules and LLMs for evaluating operation validity. Specifically, we manually implement the following rules: ① *syntactic checks* to enforce schema consistency (e.g., applying `Filter` only to attributes with well-defined domains), ② *semantic checks* to guarantee the semantic compatibility for operations, *e.g.*, a candidate answer set with movies is invalid when performing set union with another set with actors. ③ *answer checks* to enforce non-empty final answer sets. Since heuristic rules cannot enumerate all possible semantics in natural language, we further prompt GPT-4o to evaluate the correctness of each operation.

With these verification mechanisms enabled, invalid queries are pruned, and the sampling process is repeated until a valid composition is obtained or the depth termination criterion is reached. This filtering step ensures that only logically consistent and interpretable questions are retained in the final dataset.

Ambiguous Question Tracking DEEPAMBIGQA explicitly models referential ambiguity, where the question includes an ambiguous name that may lead to different reasoning paths for answering the query. To construct such questions, we first construct a set of ambiguous names, along with their linked entities, and initiate DEEPAMBIGQAGEN from these ambiguous names. Specifically, we deduplicate the alias name and initialize a branch per candidate entity. For example, the name "Nolan" yields two candidates in movie domain: *Christopher Nolan* and *Jonathan Nolan.* All subsequent operations are lifted *per branch*: we apply the same step independently to each

candidate-restricted state, keep only branches that pass `IsValid` and remain non-empty, and maintain per-branch sets of *unique* entities throughout execution. Branches that violate typing, collapse to emptiness, or lose semantic coherence are removed immediately.

At termination we enforce ambiguity-preserving constraints for these questions: ① *multi-branch*, more than two reasoning paths must survive; ② *answer divergence*, final answer sets must differ for at least two surviving branches with different key entities; and ③ *multi-hop*, each surviving path must exceed one hop. The final answer is constructed as the aggregate over all surviving branches, *i.e.*, the full answer set with all paths considered.

4 DEEPAMBIGQA DATASET

In this section, we present more details of DEEPAMBIGQA dataset, generated by the proposed DEEPAMBIGQAGEN algorithm on a Wikipedia snapshot[1]. Specifically, Section 4.1 details the data collection procedure, Section 4.2 lists the key statistics of DEEPAMBIGQA dataset, followed by Section **??** that presents the complexity and naturalness analysis of the synthesized questions.

4.1 DEEPAMBIGQA COLLECTION

In this paper, we employ the Wikipedia snapshot dated *2025-0820*[2] as the source data to build DEEPAMBIGQA. The wiki pages are aligned to corresponding wikidata annotations to obtain the knowledge graph with entities and relation annotations for query synthesis.

The DEEPAMBIGQA dataset spans a diverse set of domains derived from Wikipedia, including *movies, music, sports, books, finance, and science*. To better capture real-world ambiguity, we construct alias groups by clustering entities that share the same surface form. For example, the name *"Nolan"* may refer to both *Christopher Nolan* and *Jonathan Nolan* in the movie domain, while *"Michael Jordan"* can denote either the basketball player in sports or the computer scientist in science. Incorporating such alias sets ensures that the synthesized questions naturally involve realistic disambiguation challenges.

Question synthesis is carried out using the bottom-up pipeline introduced in Section 3, augmented with LLM-based validation and realization. In particular, GPT-4o-mini serves as a lightweight validator for checking intermediate semantic correctness and coherence of candidate query states, while GPT-4o transforms the synthesized structured queries into fluent, natural-language questions. Additional examples of domain coverage and ambiguous alias groups are provided in Appendix A.1.

4.2 DATASET STATISTICS

Table 3 summarizes the main characteristics of the DEEPAMBIGQA dataset, divided by the major source of ambiguity: *referential ambiguity* in the original query and *aggregation ambiguity* in the reasoning process. For each category, we report the average question length, answer set size, reasoning depth, and number of entities involved, which together reflect the overall complexity of the questions.

Figure 3: Key data statistics of DEEPAMBIGQA. update number.

Referential ambiguous questions:	1493
- Avg. # question tokens	23.8
- Avg. # answer set size	8.4
- Avg. # reasoning steps	7.4
- Avg. # entities involved	39.4
- Avg. # reasoning branches	4.7
Aggregation ambiguous questions:	2127
- Avg. # question tokens	19.6
- Avg. # answer set size	12.1
- Avg. # reasoning steps	6.8
- Avg. # entities involved	27.5

Referential ambiguity questions (1,493 in total) are generally longer, averaging 23.8 tokens per question. They also involve a larger reasoning space, with an average of 39.4 entities and 4.7 reasoning branches, indicating that the model must compare and resolve multiple candidate entities. Aggregation ambiguity questions (2,127 in total), by contrast, are shorter on average (19.6 tokens), but often return larger answer sets (12.1 on average). They also require multi-step reasoning (6.8 steps), highlighting the challenge of interpreting numerical or logical constraints correctly.

[1] https://en.wikipedia.org/wiki/Wikipedia:Database_download

[2] https://dumps.wikimedia.org/enwiki/20250820/

Overall, referential ambiguity emphasizes the difficulty of distinguishing between entities with similar names, while aggregation ambiguity stresses the handling of operators such as counts, comparisons, and groupings. These two forms of ambiguity thus represent complementary challenges for question answering systems. A more detailed breakdown by domain and reasoning depth is provided in Appendix A.1.

4.3 Dataset Analysis

We further conduct a human study to better assess the quality of DeepAmbigQA. The primary objective is to evaluate whether the synthesized questions are natural in phrasing and realistic in real-world and whether their corresponding answers are correct and faithful, ensuring that the dataset reflects realistic user queries while maintaining logical soundness. For each annotation task, we recruit 3 human annotators and report the average. To reduce the annotation cost, we randomly select 50 samples from each domain for both ambiguity types of questions. In total, the human study involves 600 unique questions.

Figure 4: Human evaluation results of DeepAmbigQA on question naturalness and answer correctness. Scores are averaged over three annotators. update number

Metric	Ref. Ambiguity	Agg. Ambiguity	Overall
Fluency			
Usefulness			
Precision			
Recall			
Ambiguity			

Human Study on Naturalness. We evaluate the naturalness of synthesized questions focusing on two key dimensions. First, *fluency* assesses whether a question is grammatically correct, clearly depicts the intended knowledge search task, and is phrased in a way that is natural and easily understandable by humans. Second, *usefulness* evaluates whether a question resembles those that might be asked by a real-world user seeking information, *i.e.*, whether it is plausible and contextually meaningful in practical settings.

For each annotation, we ask human annotators to rate on a 5-point Likert scale to evaluate these dimensions. We also compute inter-annotator agreement using Cohen's κ. As shown in Table 4, questions achieve high scores in both fluency (number on average) and usefulness (number on average). Inter-annotator agreement is strong, with a Cohen's κ of number overall, demonstrating that judgments are consistent across annotators. These results indicate that the majority of questions are not only well-formed linguistically but also realistic and valuable as information-seeking queries.

Human Study on Correctness. We further evaluate the correctness of the answer along three dimensions. First, *answer completeness (recall)* measures whether the returned answers exhaustively cover all gold answers that are valid for the given query. Second, *answer precision (precision)* assesses whether the listed answers are truly correct with respect to the intended interpretation, without introducing spurious items. Third, for referential ambiguity questions, we include an additional criterion of *ambiguity awareness*, which evaluates whether the answer set appropriately considers all meaningful answers corresponding to different but valid entity interpretations.

Annotators verify these criteria by inspecting per-branch provenance for referential ambiguity questions and by checking all admissible operator interpretations for aggregation ambiguity questions. As summarized in Table 4, precision is strong overall (number%). Recall is also high (number% overall), though aggregation questions show minor drops due to operator-based reasoning challenges. Ambiguity awareness is likewise well preserved, with annotators confirming that most referential cases account for multiple legitimate interpretations. Together, these results demonstrate that DeepAmbigQA provides answers that are not only complete and correct but also sensitive to the nuances of ambiguity inherent in real-world queries.

5 Experiment

In this section, we evaluate various families of frontier LLMs on the constructed DeepAmbigQA dataset and conduct an in-depth analysis of how different components in the retrieval process affect performance. Specifically, Section 5.1 presents the evaluation setup and reports the performance of various LLMs, along with a detailed breakdown of failure cases 5.2. Next, Section 5.3 examines

Table 2: Evaluation results on DEEPAMBIGQA. We report Precision (P), Recall (R), Exact Match (EM), and the average number of queries (Q) for both referential ambiguity and aggregation ambiguity queries.

Model	Referential Ambiguity				Aggregation Ambiguity			
	P	R	EM	Q	P	R	EM	Q
GT	–	–	–	4.8	–	–	–	5.43
GPT-4o	85.2	82.1	78.5	3.4	83.0	80.4	76.2	3.2
GPT-4o-mini	83.6	80.2	76.4	3.2	81.0	77.9	73.8	3.1
GPT-5-mini	84.5	81.0	77.1	3.5	82.3	78.8	74.7	3.3
GPT-5	86.1	83.0	79.2	3.3	84.0	81.5	77.3	3.1
Qwen2.5-3B	–	–	–	–	–	–	–	–
Qwen2.5-8B	–	–	–	–	–	–	–	–
DeepSeek-R1-distill-1.5B	–	–	–	–	–	–	–	–
DeepSeek-R1-distill-7B	–	–	–	–	–	–	–	–
Qwen3-4B	–	–	–	–	–	–	–	–
Qwen3-8B	–	–	–	–	–	–	–	–
Gemini-1.5-Flash	–	–	–	–	–	–	–	–
Gemini-1.5-Pro	–	–	–	–	–	–	–	–

the impact of individual system components, such as the additional module, the retriever, and chunk size, on the overall performance.

5.1 EXPERIMENT SETTING

We evaluate 10 large language models (LLMs) spanning both closed-source and open-source families, including the GPT series (OpenAI, 2023), Gemini series (Gemini Team, 2024), Qwen2.5 series (Team, 2025a), Qwen3 series (Team, 2025b), and the DeepSeek-R1 distilled series (DeepSeek-AI, 2025). All models are prompted with the same template to generate answers in JSON dictionary format (see Appendix A.4).

For models equipped with internal tool-use capabilities supporting retrieval APIs, we instruct them to query the local wiki corpus via the provided API. For models without tool-use functionality, we implement equivalent retrieval handling as an agentic workflow. The wiki corpus is preprocessed following the FlashRAG (Jin et al., 2024) pipeline, which extracts and chunks wiki pages. Unless otherwise specified, we employ e5-base [3] as the embedding model for indexing. Additional implementation details are presented in Appendix A.

Since answers in DEEPAMBIGQA are always represented as sets of entities, we report precision, recall, and exact match (EM) for each query. We also include the average number of queries issued by each LLM during evaluation.

5.2 EXPERIMENT RESULTS

LLM Performance on DEEPAMBIGQA Table 2 presents the overall performance of various LLMs on DEEPAMBIGQA. We highlight following observations: wait for full results for analysis

Performance Breakdown To better understand the system performance, we break down results based on two dimensions: (i) reasoning depth, *i.e.*, the number of reasoning steps required, and (ii) referential ambiguity, *i.e.*, the number of unique branches involved in referentially ambiguous ques-

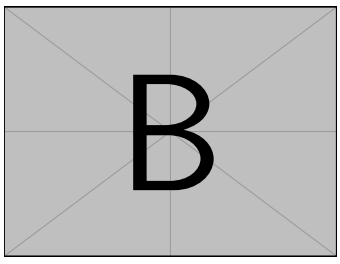

Figure 5: Performance breakdown by reasoning depth and referential ambiguity.

[3] https://huggingface.co/intfloat/e5-base-v2

Table 3: Impact of query rewriting and evidence extraction on DEEPAMBIGQA performance.

Model	Baseline			+ Query Rewrite			+ Evidence Extraction			+ Knowledge Graph		
	P	R	EM	P	R	EM	P	R	EM	P	R	EM
GPT-4o												
GPT-4o-mini												
GPT-5-mini												
GPT-5												
Qwen2.5-3B												
Qwen2.5-8B												
DeepSeek-R1-distill-1.5B												
DeepSeek-R1-distill-7B												
Qwen3-4B												
Qwen3-8B												
Gemini-1.5-Flash												
Gemini-1.5-Pro												

tions. Figure **??** presents the result. We highlight following observations: wait for result

Failure Breakdown and Analysis To better understand error patterns, we categorize failures into (i) missing relevant entities, (ii) hallucinating incorrect entities, and (iii) partial coverage (correct but incomplete sets). Figure 6 illustrates the distribution of these failure modes across models.

5.3 ADDITIONAL ANALYSIS

We further study how different components affect LLM performance on DEEPAMBIGQA. To accommodate this, we design two additional modules: ❶ *Query rewriting*, which prompts an LLM to rewrite the original query to disambiguate entities with similar names. ❷ *Evidence extraction*, which prompts an LLM to remove unnecessary information from retrieved text and retain only relevant spans, thereby mitigating aggregation errors.

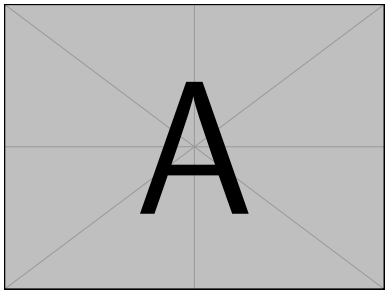

Figure 6: Breakdown of failure cases across LLM families.

Effect of Query Rewriting Query rewriting is designed to disambiguate entities with similar or overlapping names, thereby reducing retrieval errors caused by ambiguity in the original query. Table **??** reports the performance before and after applying query rewriting across several LLMs.

Effect of Evidence Extraction Evidence extraction filters retrieved passages to retain only the content relevant to the query, which can reduce noise and improve aggregation. Table 3 compares performance across models with and without evidence extraction.

Effect of Knowledge-graph Information In addition to text-based retrieval, we investigate whether incorporating structured knowledge from a knowledge graph can improve performance. Specifically, we prompt LLMs to directly generate SPARQL queries to solve questions. This approach allows models to retrieve precise entity relations without relying solely on unstructured passages. Table 3 compares baseline retrieval with the knowledge-graph–augmented setting.

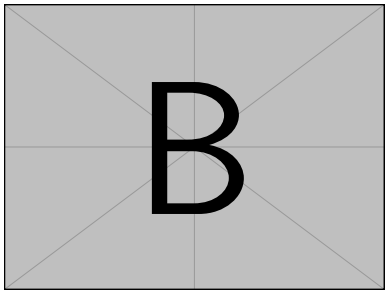

Figure 7: Effect of retriever types for system performance.

Effect of Retriever Given the central role of retrieval in the system pipeline, we examine the impact of different retrievers. Specifically, we evaluate sparse retrieval (*bm25*), dense retrieval methods (*e5-base* and *qwen3-0.6B embedding*), and an oracle configuration in which the ground-truth wiki pages are supplied as context. Figure 7 presents the results wait for result

6 Conclusion

In this paper, we introduced DeepAmbigQA, a benchmark that isolates ambiguity as a central challenge in deep-research question answering. By constructing questions that require careful referential disambiguation and evidence aggregation, DeepAmbigQA exposes systematic weaknesses in current LLMs, particularly incomplete retrieval and erroneous entity resolution. Our results underscore the need for advances in retrieval and reasoning methods, and we position DeepAmbigQA as a resource to drive progress toward more reliable deep-research QA.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *SIGIR*, 2019.

Mohammad Aliannejadi et al. Convai3: Generating clarifying questions for open-domain dialogue systems (ClariQ). In *SCAI Workshop at EMNLP (shared task report)*, 2020.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. Open-domain question answering goes conversational via question rewriting. In *NAACL*, 2021.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection, 2023.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, et al. Improving language models by retrieving from trillions of tokens. In *ICML*. PMLR, 2022.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *ACL*, 2017.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. GENRE: Autoregressive entity retrieval. In *ICLR*, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. Can you unpack that? learning to rewrite questions-in-context. In *EMNLP*, 2019.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. IIRC: A dataset of incomplete information reading comprehension questions. In *EMNLP*, 2020.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *ACL*, 2023.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. 2024. URL https://arxiv.org/abs/2403.05530.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *TACL*, 2021.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *ICML*, 2020.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps (2WikiMultiHopQA). In *COLING*, 2020.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *ICLR*, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *JMLR*, 24(337):1–43, 2023.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576, 2024. doi: 10.48550/ARXIV.2405.13576. URL https://doi.org/10.48550/arXiv.2405.13576.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

Tushar Khot, Ashish Sabharwal, and Peter Clark. QASC: A dataset for question answering via sentence composition. In *AAAI*, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, et al. Natural questions: A benchmark for question answering research. *TACL*, 2019.

Matthew Lamm, Jay DeYoung, et al. QED: A framework and dataset for explanations in question answering. *TACL*, 2021.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. A survey of large language models in finance (finllms). *arXiv preprint arXiv:2402.02315*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, 2020.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. ELQ: Efficient one-pass end-to-end entity linking for questions. In *EMNLP*, 2020.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*, 2020.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback, 2021.

OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Laurel Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *CIDR*, 2021.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *NAACL*, 2021.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, Jacob Steinhardt, and Percy Liang. Measuring and narrowing the compositionality gap in language models, 2022.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. Open-retrieval conversational question answering. In *SIGIR*, 2020.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, 2018.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *EMNLP*, 2022.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *NAACL*, 2018.

Qwen Team. Qwen2.5 technical report. 2025a. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3 technical report. 2025b. URL https://arxiv.org/abs/2505.09388.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. In *EMNLP*, 2022.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. In *TACL/ICLR*, 2018.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, 2020.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Answering complex questions with multi-hop dense retrieval. In *ICLR*, 2021.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.

A Implementation Detail

A.1 DeepAmbigQA Generation Algorithm

A.2 DeepAmbigQA Details

A.3 Evaluation Setting

A.4 Prompts

In this section, we list all prompts employed in this work.

B Additional Examples