WildGraphBench: Benchmarking GraphRAG with Wild-Source Corpora

Pengyu Wang[1]* **Benfeng Xu**[1,2]† **Licheng Zhang**[1]‡ **Shaohan Wang**[1]
Mingxuan Du[1] **Chiwei Zhu**[1] **Zhendong Mao**[1]
[1]University of Science and Technology of China, Hefei, China
[2]Metastone Technology, Beijing, China
{wangpengyu, benfeng, zlczlc}@mail.ustc.edu.cn, zdmao@ustc.edu.cn

Abstract

Graph-based Retrieval-Augmented Generation (GraphRAG) organizes external knowledge as a hierarchical graph, enabling efficient retrieval and aggregation of scattered evidence across multiple documents. However, many existing benchmarks for GraphRAG rely on short, curated passages as external knowledge, failing to adequately evaluate systems in realistic settings involving long contexts and large-scale heterogeneous documents. To bridge this gap, we introduce *WildGraphBench*, a benchmark designed to assess GraphRAG performance in the wild. We leverage Wikipedia's unique structure, where cohesive narratives are grounded in long and heterogeneous external reference documents, to construct a benchmark reflecting real-world scenarios. Specifically, we sample articles across 12 top-level topics, using their external references as the retrieval corpus and citation-linked statements as ground truth, resulting in 1,100 questions spanning three levels of complexity: single-fact QA, multi-fact QA, and section-level summarization. Experiments across multiple baselines reveal that current GraphRAG pipelines help on multi-fact aggregation when evidence comes from a moderate number of sources, but this aggregation paradigm may overemphasize high-level statements at the expense of fine-grained details, leading to weaker performance on summarization tasks.

1 Introduction

Retrieval-augmented generation (RAG) grounds LLM outputs by retrieving evidence from an external corpus(Lewis et al., 2021), typically via dense passage retrieval (Karpukhin et al., 2020) or sparse methods like BM25, but it may struggle when scattered evidence must be extracted from multiple documents and integrated into a coherent answer. Recently, Graph-based RAG (GraphRAG) (Peng et al., 2024) has gained increasing attention as a paradigm that builds a graph over documents or chunks and performs graph-guided expansion and aggregation for multi-document evidence assembly and long-context reasoning (Zhang et al., 2025).

Many GraphRAG methods have been proposed, exploring different graph constructions and retrieval strategies. Microsoft GraphRAG(Edge et al., 2025) builds document-level graphs and supports local-to-global aggregation for query-focused summarization, while LightRAG(Guo et al., 2025) improves evidence coverage by coupling an entity–relation graph with vector retrieval and multi-stage expansion. In addition, some works attempt to improve practicality. Fast-GraphRAG(CircleMind-AI, 2024) focuses on efficiency and adopts a lightweight graph retrieval pipeline to reduce indexing and querying costs. HippoRAG2(Gutiérrez et al., 2025) further extends evidence access by introducing an external knowledge graph and filtering mechanism. LinearRAG(Zhuang et al., 2025) enhances the practicality by adopting a lightweight hierarchy and propagation-style ranking for scalable indexing and retrieval. These strategies enable structured evidence expansion and aggregation, allowing LLMs to address more complex queries.

However, current GraphRAG benchmarks still rely on short, curated passages. As a result, retrieval and generation in long-context settings with large-scale, heterogeneous document collections remain under-tested. Yet this setting is central to real-world applications and is also where GraphRAG is expected to be most beneficial; moreover, recent work has shown that even standard flat-RAG techniques can be brittle when confronted with irrelevant or noisy context (Yoran et al., 2024). In addition, many benchmarks make the task closer to lookup-and-stitch: once a few pre-trimmed passages are retrieved, a system can answer by simple concatenation or light paraphrasing rather than true

*Work done during the internship at Metastone.
†Project lead.
‡Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 13875–13890

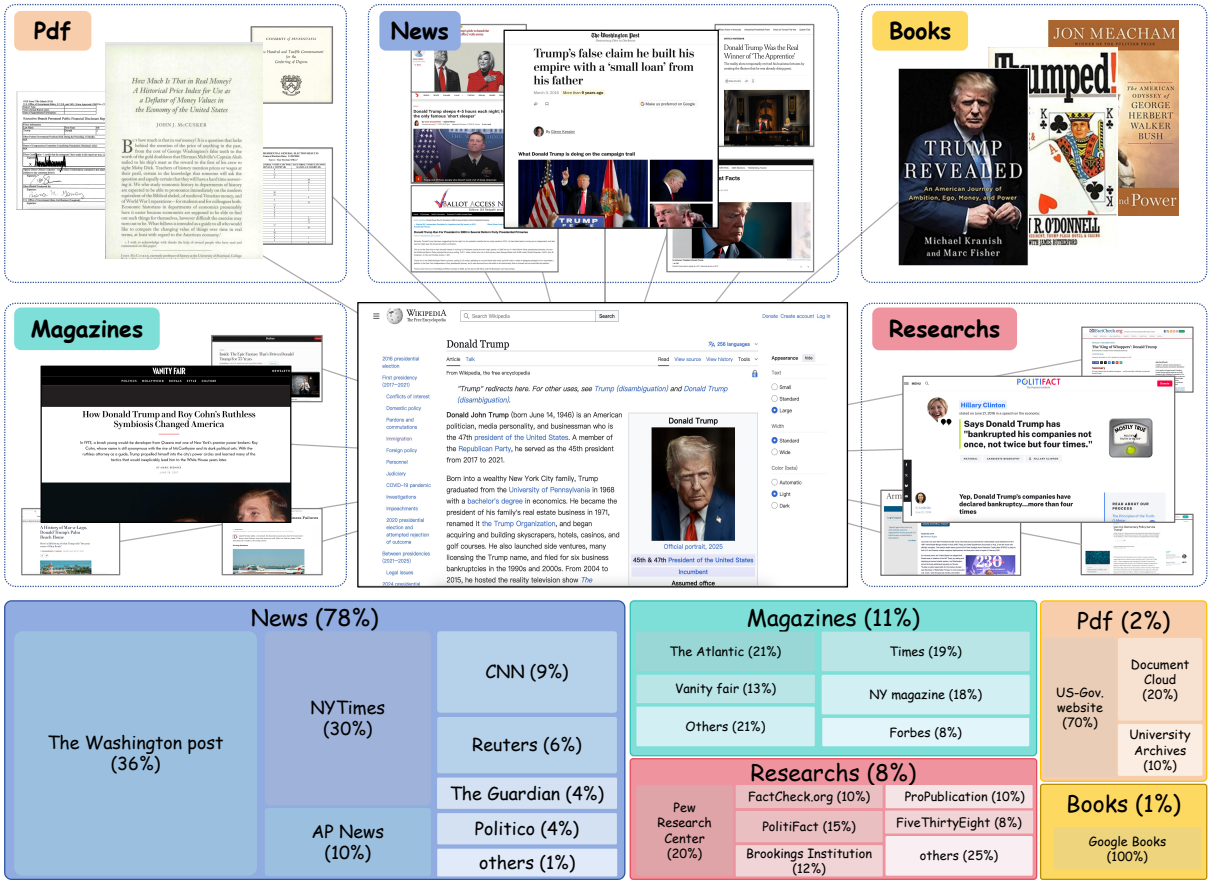

Figure 1: Why we use Wikipedia references as wild evidence. Wikipedia articles are concise summaries with citation-linked statements. The linked reference pages are often long, noisy, and heterogeneous (e.g., news sites, blogs, PDFs, and public reports). This mismatch makes evidence retrieval and verification harder.

multi-document aggregation. Traditional datasets such as HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MultiHop-RAG (Tang and Yang, 2024) typically follow this assumption. Even some recent benchmarks extend to longer domain documents, e.g., UltraDomain (Qian et al., 2025), they still assume cleaner document boundaries and less heterogeneous sources than wild corpora. For GraphRAG specifically, a few dedicated benchmarks have recently been proposed (Xiao et al., 2025; Xiang et al., 2025), providing controlled protocols and corpora; however, their corpora remain more structured and less heterogeneous than in-the-wild settings. This gap calls for tasks that test reliable aggregation under long contexts and heterogeneous, uncurated sources.

To bridge this gap, we introduce *WildGraphBench*, a benchmark that targets the in-the-wild scenario of GraphRAG, where a system is expected not only to retrieve evidence from long, heterogeneous corpora, but also to synthesize an answer whose correctness depends on assembling scattered support across multiple sources rather than a handful of pre-trimmed passages. We instantiate this setting using Wikipedia: each article provides a concise entry with citation-linked statements, while its external reference pages form a long, heterogeneous web corpus (Figure 1). We therefore sample Wikipedia articles from 12 top-level topics, using each article's reference pages as the retrieval corpus and treating citation-linked Wikipedia statements as ground-truth facts. We create 1,100+ questions in three types, as shown in Figure 2, spanning single-fact lookup, multi-fact evidence aggregation, and section-level summarization, which together stress the spectrum from precise retrieval to broad factual coverage. Our contributions are as follows:

- We construct *WildGraphBench*, a dataset based on Wikipedia references that reflects real-world noise and complexity.
- We design three question types: single-fact, multi-fact, and section-level summary. Then we introduce a statement-grounded evaluation

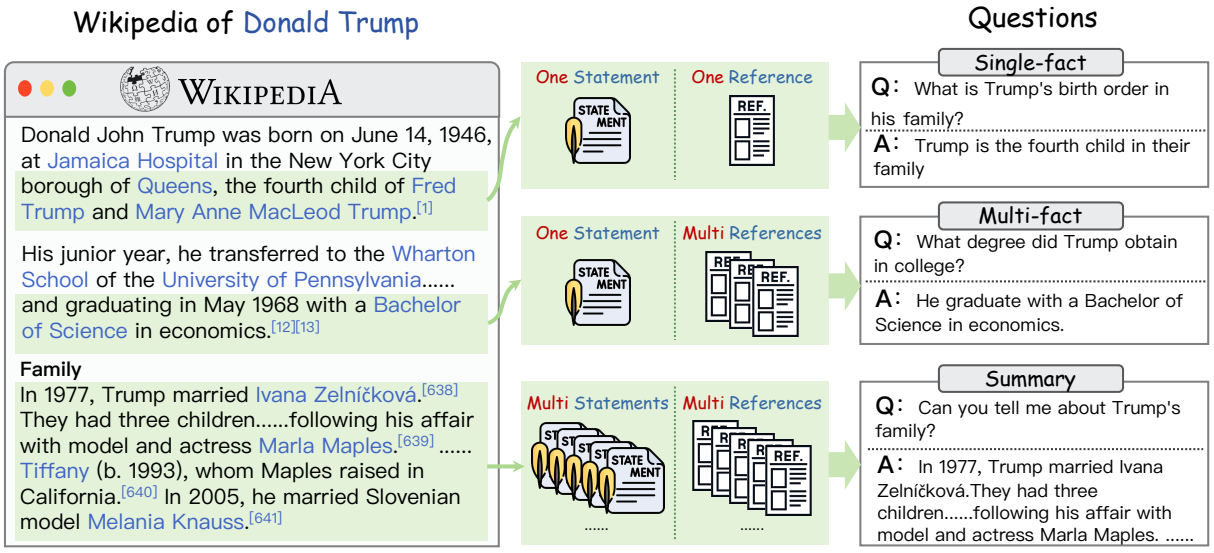

Figure 2: Example instances in *WildGraphBench*: (1) single-fact questions grounded by a single gold statement and one reference, (2) multi-fact questions requiring evidence aggregation across multiple statements/references, and (3) section-level summary questions evaluated at the statement level.

method for single-fact, multi-fact, and summary questions.

- Experiments on multiple methods show that GraphRAG improves multi-fact aggregation but struggles with broad summary tasks.

2 Related Work

Retrieval-Augmented Generation. Retrieval-augmented generation (RAG) answers a query by retrieving related text from an external corpus, then generating based on that text (Lewis et al., 2021). A common setup uses a retriever (e.g., BM25 or dense passage retrieval (Karpukhin et al., 2020)) plus an LLM reader (Robertson and Zaragoza, 2009; Chen et al., 2017). Prior work shows retrieval can reduce hallucination compared to direct generation, especially when evidence is needed (Shuster et al., 2021), though recent studies also highlight the challenge of robustness to irrelevant context (Yoran et al., 2024).

Graph Retrieval-Augmented Generation. Graph retrieval-augmented generation (GraphRAG) builds a graph over chunks or documents and retrieves evidence through graph operations, aiming to capture complex dependencies that flat retrieval might miss. Microsoft GraphRAG (Edge et al., 2025) introduces a modular pipeline that uses community detection algorithms to generate hierarchical summaries, supporting both local entity queries and global thematic answering. LightRAG (Guo et al., 2025) constructs a dual-level entity–relation graph and couples it with vector retrieval, allowing for multi-stage evidence expansion across low-level entities and high-level concepts. Focusing on efficiency, Fast-GraphRAG (CircleMind-AI, 2024) adopts a lightweight design with optimized indexing strategies to significantly reduce the computational overhead of graph maintenance. HippoRAG2 (Gutiérrez et al., 2025) draws inspiration from the human brain, introducing an external knowledge graph and a memory-style retrieval mechanism powered by Personalized PageRank (PPR) to filter noise and discover multi-hop paths. Furthermore, LinearRAG (Zhuang et al., 2025) targets large-scale scalability by utilizing a linear-complexity propagation ranking method, overcoming the bottleneck of traditional graph algorithms. These methods collectively demonstrate that structured graph traversal can significantly enhance evidence aggregation compared to flat baselines.

Benchmarks for RAG and GraphRAG. Several benchmarks study retrieval and multi-hop reasoning under different evidence settings. HotpotQA(Yang et al., 2018), 2WikiMultiHopQA(Ho et al., 2020), and MultiHop-RAG(Tang and Yang, 2024) focus on multi-hop question answering. UltraDomain(Qian et al., 2025) evaluates retrieval and generation over longer domain corpora.

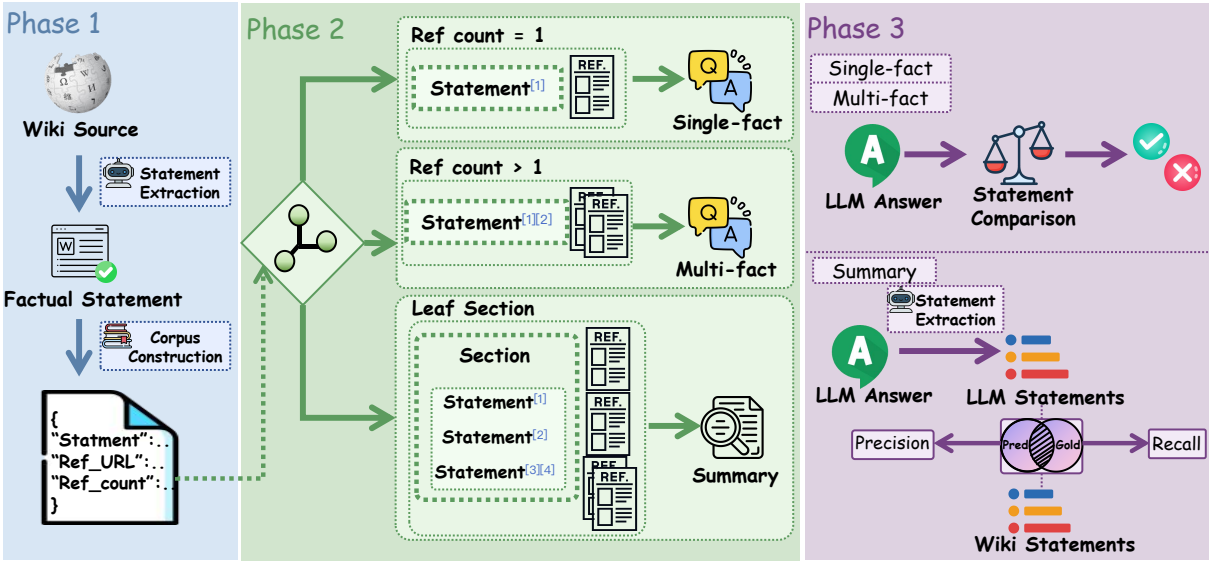

Figure 3: Three-phase workflow of *WildGraphBench* after data collection. Phase 1:citation-aware statement extraction, producing the Wikipedia gold corpus. Phase 2: design single-fact, multi-fact, and section-level summary questions. Phase 3: evaluate with statement-grounded accuracy and statement-level precision/recall/F1.

GraphRAG-Bench(Xiao et al., 2025) provides corpora and protocols tailored to graph-based retrieval. Additionally, Xiang et al. (Xiang et al., 2025) systematically analyze the scenarios where graph structures provide a clear benefit over flat retrieval. Despite these advances, there remains a gap for a benchmark that simultaneously stresses long-context processing, noise robustness, and multi-document aggregation in a wild, unstructured web setting.

3 WildGraphBench

In this section, we describe how *WildGraphBench* is constructed. As shown in Figure 3, the framework has three phases. First, we collect reference pages and extract citation-linked statements from Wikipedia leaf sections, producing the Wikipedia gold corpus. Based on the extracted corpus, we then build three types of questions: single-fact, multi-fact, and section-level summary. Finally, we introduce a statement-grounded evaluation method.

3.1 Data Collection

We start from 12 high-level Wikipedia topics[1] and within each topic select articles with a large number of references, as these articles tend to have dense and diverse citation structures. For each article, we collect all reference URLs and fetch the original web pages using jina.ai.[2]. Crucially, if the original page fails but an archive exists, we use the archived page to ensure data completeness. We keep the raw page text, including boilerplate and noise, to simulate the wild retrieval environment.

3.2 Statement Extraction

We use citation-linked Wikipedia statements as the ground-truth factual units for evaluation. To extract statements, we first split each Wikipedia article into leaf sections using a simple regex parser over Wiki markup. Each leaf section has a section path (e.g., *Donald Trump* > *Political Career* > *Impeachments*). In each leaf section, we identify sentences containing citation markers. For each sentence, an LLM rewrites it into a clean factual statement by removing footnote markers and fixing local coreference issues. We also parse the Wiki markup to retrieve the exact reference URLs.

3.3 Wikipedia Gold Corpus Construction

We align the references by matching Wikipedia reference URLs to the crawled pages. If a cited sentence is missing any referenced page, we drop it to ensure quality. The resulting Wikipedia gold corpus is organized at the granularity of leaf sections, where each section stores a list of triples:

$$\mathcal{T} = (statement, ref\_urls, ref\_count) \quad (1)$$

[1]Source: Wikipedia:Contents. It lists 13 top-level topics: Culture, Geography, Health, History, Human activities, Mathematics, Nature, People, Philosophy, Religion, Society, Technology, and Reference. We drop *Reference*. It mainly describes how to write and cite Wikipedia pages, rather than a content domain.

[2]https://jina.ai

Question Type	Count
Single-Fact	667
Multi-Fact	191
Summary	339
Total	**1,197**

Table 1: Statistics of Question Types in *WildGraph-Bench*. The dataset consists of 1,197 questions distributed across three distinct categories.

where $statement$ is the normalized factual statement, ref_urls is the set of reference URLs associated with the original cited sentence, and ref_count is the number of references. This corpus serves as the authoritative source of gold statements for evaluation.

3.4 Question Design

We design three types of questions on top of the Wikipedia gold corpus. The reference count ref_count attached to each Wikipedia triple determines whether it is used for a single-fact or multi-fact question, while leaf sections themselves serve as the basis for summary questions (Table 1).

Single-fact questions. If a triple has $ref\_count = 1$, we use it to generate a single-fact question. Given the article title, the section path, the original source sentence, and the cleaned statement, we prompt an LLM to write a question whose answer is exactly that statement (up to minor paraphrasing). The prompt specifically encourages the model to include multiple constraints (e.g., entity, time, and location) and discourages copying long spans from the statement, ensuring that the question is non-trivial but still tightly aligned with the gold statement and its supporting evidence.

Multi-fact questions. If a triple has $ref\_count \geq 2$, we treat it as a candidate for a multi-fact question. Intuitively, these statements are supported by multiple references and often describe relationships that span several sources. We again condition an LLM on the article title, section path, and statement to generate a question that requires recovering that statement. In addition, we enforce a strict multi-reference check: for each such triple, we ask an LLM judge whether any single reference alone is sufficient to support all key facts in the statement, and only keep those triples for which at least two references are jointly required. This yields questions that genuinely test a model's ability to aggregate evidence from multiple documents.

Section-level summary questions. For summary questions, we operate at the level of leaf sections. For a given leaf section, we collect all valid triples under that section and deduplicate their statements to obtain the gold statement set S^*. We then prompt an LLM to generate a natural information-seeking question based on the article title and the section path, but not on the section text itself, so that the question is phrased independently of the exact wording of Wikipedia. The expected answer to such a question is the set S^*, i.e., the factual content of the leaf section. During evaluation, systems are required to retrieve and summarize information under noisy, long-context evidence conditions.

3.5 Evaluation Metrics

We evaluate systems differently for single-fact/multi-fact questions and for summary questions, but in all cases the gold answer is defined at the level of factual statements.

Single-fact and multi-fact accuracy. For single-fact and multi-fact questions, each instance is associated with exactly one gold statement s^*. Given a system answer $\hat{a}$ and the relevant evidence, an LLM judge decides whether $\hat{a}$ is factually equivalent to s^* under the evidence. We assign a score of 1 if the answer is correct and 0 otherwise, and report accuracy separately for single-fact and multi-fact questions.

Statement-level score for summary. For summary questions, the gold answer is the statement set S^* extracted from the corresponding leaf section. Given a system output, we run a statement extractor to obtain a set of predicted statements $\hat{S} = \{\hat{s}_1, \ldots, \hat{s}_k\}$. We then define a binary matching function $Match(s, \hat{s}) \in \{0, 1\}$, which returns 1 if $\hat{s}$ is a correct paraphrase of s (i.e., it conveys the same fact), and 0 otherwise. Using this match, we compute statement-level recall and precision as:

$$\text{Recall} = \frac{1}{|S^*|} \sum_{s \in S^*} \max_{\hat{s} \in \hat{S}} Match(s, \hat{s}) \quad (2)$$

$$\text{Precision} = \frac{1}{|\hat{S}|} \sum_{\hat{s} \in \hat{S}} \max_{s \in S^*} Match(s, \hat{s}) \quad (3)$$

The F1 score is the harmonic mean of precision and recall. This metric directly measures factual

Method	Question Answering			Summary		
	Avg. Acc.	Single-fact Acc.	Multi-fact Acc.	Recall	Precision	F1
NaiveRAG	59.79	66.87	35.08	**13.54**	19.07	**15.84**
BM25	36.83	41.38	20.94	9.38	19.46	12.66
Fast-GraphRAG	33.56	35.83	25.65	6.81	23.48	10.56
HippoRAG2	**64.33**	**71.51**	39.27	11.15	16.76	13.39
Microsoft GraphRAG(local)	38.23	39.43	34.03	9.82	12.64	11.05
Microsoft GraphRAG(global)	54.54	56.52	**47.64**	12.66	15.13	13.78
LightRAG(hybrid)	56.76	61.32	40.84	12.44	17.7	14.61
LinearRAG	44.87	47.53	35.6	5.81	**29.2**	9.69

Table 2: Main results on *WildGraphBench*. While graph-based methods show clear advantages on multi-fact questions requiring aggregation, flat baselines like NaiveRAG remain competitive on single-fact retrieval and achieve higher recall on summary tasks due to broader context coverage.

coverage (recall) and hallucination rate (precision), while remaining robust to minor paraphrasing.

4 Experiments

4.1 Experimental Settings

Implementation Details For methods requiring pre-chunking, we segment documents with chunk size 1200 tokens and overlap 100 tokens. At retrieval time, we set `top_k` $= 5$ for single-fact and multi-fact questions, and `top_k` $= 10$ for summary questions. To align generation and graph construction across systems, we use `gpt-4o-mini` as the default model for graph construction and answering. For evaluation, we use `gpt-5-mini` as the LLM judge to score single-fact and multi-fact accuracy and to compute statement-level precision/recall/F1 for summary questions.

Evaluated Methods We evaluate *WildGraphBench* on representative flat-RAG and GraphRAG-style baselines. For flat-RAG, we include NaiveRAG (Lewis et al., 2021) and BM25 (Robertson and Zaragoza, 2009). For GraphRAG-style methods, we evaluate Fast-GraphRAG (CircleMind-AI, 2024), Microsoft GraphRAG (local/global) (Edge et al., 2025), LightRAG (hybrid) (Guo et al., 2025), LinearRAG (Zhuang et al., 2025), and HippoRAG2 (Gutiérrez et al., 2025).

4.2 Main Results

Table 2 reports the overall performance on *WildGraphBench* and the `people` subset. We observe a consistent pattern across both parts.

On the relatively simple **single-fact** questions, flat retrieval-augmented baselines remain highly competitive. In particular, NaiveRAG achieves strong accuracy, outperforming most graph-based variants on the Novel Dataset; among the GraphRAG-style methods, only HippoRAG2 attains a higher single-fact accuracy (71.51 vs. 66.87), suggesting that graph retrieval does not automatically translate into gains when the answer can often be supported by a single salient chunk. BM25 is also competitive, and in some cases surpasses several graph methods, indicating that keyword matching remains a strong prior for straightforward fact lookup.

The advantage of graph-based retrieval becomes more visible on harder tasks. For **multi-fact** questions, which require aggregating evidence from multiple references, Microsoft GraphRAG(global) achieves the best accuracy(47.64), and several graph variants are comparable to or better than NaiveRAG and BM25. This implies that structured traversal / global context aggregation can help when evidence is scattered and must be combined, while a purely flat top-k pipeline is more likely to miss complementary pieces of information.

For **summary** questions, all methods obtain low statement-level scores, highlighting the difficulty of reconstructing a leaf section's factual content from long, noisy evidence. Notably, **NaiveRAG achieves the highest recall and the best F1** on our Dataset (Table 2). A plausible explanation is that summary questions demand *broad coverage*: retrieving a wider variety of raw evidence chunks can directly increase the chance that the generator sees more gold facts, improving recall and thus F1. In contrast, many GraphRAG-style systems introduce additional bottlenecks—entity/relation extraction, graph sparsification, neighborhood summarization, and traversal budgets—which may degrade under web noise and long contexts. When graph construction is imperfect (missing entities/edges) or

Method	Question Answering			Summary		
	Ave. Acc.	Single-fact Acc.	Multi-fact Acc.	Recall	Precision	F1
NaiveRAG	65.82	76.62	28.12	**10.48**	15.29	**8.03**
BM25	65.2	74.03	34.38	5.74	16.98	5.03
Fast-GraphRAG	30.43	33.77	18.75	1.48	**22.83**	1.62
HippoRAG2	64.89	72.73	37.5	7.63	15.69	6.14
Microsoft GraphRAG(local)	35.16	38.96	21.88	4.59	9.17	2.98
Microsoft GraphRAG(global)	56.81	62.34	37.5	5.52	14.13	5.41
LightRAG(hybrid)	**74.42**	**80.52**	**53.12**	5.56	15.69	4.73
LinearRAG	45.26	51.95	21.88	1.52	22.51	1.69
Human performance	85.66	89.61	71.88	38.59	12.62	15.3

Table 3: Results on the `people` subset of *WildGraphBench*, compared with human performance.

when traversal/summarization budgets are limited, these methods can fail to *scale* their evidence gathering to the breadth required by section-level summaries, resulting in lower recall and weaker overall F1 even if they sometimes improve precision via filtering. This suggests that scaling GraphRAG to summary-style tasks under wild evidence requires more robust graph construction and higher-capacity aggregation, rather than relying on graph structure alone.

Overall, these results indicate that **GraphRAG is not always advantageous on easy questions**—it can be more expensive than NaiveRAG or BM25 without clear gains for single-fact lookup. While GraphRAG shows promising improvements on multi-fact aggregation, summary questions remain challenging under noisy long-context evidence, and method design must carefully balance *coverage* (recall) versus *filtering* (precision) under a fixed retrieval/compute budget.

4.3 Graph Analysis

Figure 4 and Table 4 compare graph connectivity across datasets under the same LightRAG construction pipeline. Following Xiang et al. (2025), a more *organized* graph should avoid excessive isolated nodes, because graph-based retrieval (e.g., traversal, community aggregation, or PPR-style propagation) relies on connectivity to reach and combine evidence beyond a single chunk. Consistent with this criterion, our graph achieves the **highest average degree** (3.11) and the **lowest proportion of isolated nodes** (0.14), indicating denser cross-document links and fewer disconnected components.

In contrast, the other datasets are structurally less favorable for graph connectivity under the same pipeline.

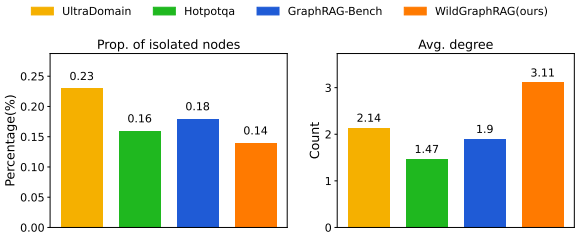

Figure 4: **Graph Quality.** We build graphs with LightRAG. Left: the fraction of isolated nodes (lower means better connectivity). Right: the average degree (higher means denser links).

Dataset	Tokens	Nodes	Degrees	Max Degree
UltraDomain	15.1M	33018	35284	195
HotpotQA	1.4M	28282	20762	50
GraphRAG-Bench	1.1M	11150	10638	155
Ours	2.4M	23940	**37246**	**967**

Table 4: Graph structural statistics across datasets constructed using the LightRAG pipeline. *WildGraphBench* exhibits a significantly higher max degree and denser connectivity, reflecting the complex, hub-centric nature of wild reference pages.

UltraDomain is constructed from curated long-context domain corpora (e.g., textbooks and domain documents), where content is organized by chapters/sections with cleaner boundaries and less repeated entity co-occurrence across documents; as a result, entity mentions are more "local", producing more low-degree or isolated nodes.

HotpotQA provides short, paragraph-level Wikipedia evidence for multi-hop QA; such contexts are comparatively compact and often concentrate on answering a specific question with a small set of supporting paragraphs, leading to weaker global entity sharing and thus a sparser graph.

GraphRAG-Bench exhibits structural sparsity

due to domain-specific characteristics across its subsets. The **Novel** subset features narrative-style text where entities and events are often distributed with strong locality (e.g., confined to specific scenes or chapters), and coreference-heavy writing dilutes explicit entity overlap during extraction. Similarly, the **Medical** subset—composed largely of biomedical guidelines from high terminological density and distinct, non-narrative structures. The lack of explicit continuity between independent medical documents isolates entities within their specific contexts, while the extraction ambiguity of specialized terms (e.g., drug names) by general-purpose LLMs further exacerbates graph fragmentation, resulting in lower average degrees and more disconnected components across the benchmark.

Finally, Table 4 highlights a striking **max-degree** gap: although the UltraDomain is much larger in total token budget than ours, our graph still exhibits a dramatically larger max degree. This suggests the presence of *hub entities* that are repeatedly linked by many distinct pages, i.e., multiple sources converge on the same entity-centric node. Such hub-and-spoke patterns make the benchmark substantially harder for retrieval and generation: systems must aggregate partially overlapping evidence from many documents and synthesize a coherent answer, directly stressing cross-document multi-source summarization and reasoning—the key capability that *WildGraphBench* aims to evaluate.

To further probe whether multi-step flat retrieval can match graph-based methods, we test two additional baselines on the Culture domain: NaiveRAG + Reranking and Query Decomposition. As shown in Table 5, Query Decomposition achieves the lowest multi-fact accuracy at 16.22%, confirming that graph structure, not multi-step retrieval, drives multi-fact gains. Adding reranking actually degrades single-fact accuracy from 67.44% to 56.98%, suggesting that standard reranking designed for clean corpora does not transfer well under wild-source noise.

Table 6 compares the indexing and query efficiency of all methods on the Culture domain. Flat baselines complete indexing in seconds, while GraphRAG methods range from 28 minutes to 21.5 hours. LLM token consumption varies from 0 for LinearRAG to 90M for LightRAG. For query latency, Fast-GraphRAG and HippoRAG2 are fastest at 5–6s, while MS GraphRAG global mode reaches 39.1s.

Method	SF Acc.	MF Acc.	Sum. F1
NaiveRAG / DPR	67.44	32.43	9.79
+ Reranking	56.98	21.62	6.61
Query Decomposition	50.00	16.22	10.01
MS GraphRAG (global)	52.33	51.35	1.80

Table 5: Additional flat RAG baselines on Culture. Graph structure, not multi-step retrieval, drives multi-fact gains.

Method	Idx Time	Nodes	Edges	MB	Avg Lat.	Idx Tok.
NaiveRAG	43s	–	–	–	2.6s	0
BM25	21s	–	–	–	2.5s	0
Fast-GraphRAG	2.3h	14.6K	35.4K	80	5.4s	17.6M
HippoRAG2	28min	49.3K	1.16M	1096	5.8s	14.9M
MS GraphRAG	49min	13.1K	24.2K	191	23.5/39.1s	60.1M
LightRAG	21.5h	23.9K	37.1K	1061	12.7s	90.0M
LinearRAG	54min	90.0K	315.8K	679	17.2s	0

Table 6: Indexing and query efficiency on the Culture domain. For MS GraphRAG, average latency is reported as local/global.

4.4 Human Performance

We recruit domain-knowledgeable annotators (graduate-level or above) and ask them to answer questions under the same evidence constraint as RAG systems. Interestingly, we observe a distinct behavior in human responses for summary tasks: human annotators tend to prioritize comprehensive coverage of key facts, attempting to include as many details as possible. While this approach sometimes results in lower precision—as models are often more conservative in their generation—the overall F1 score for human performance remains high (e.g., 15.30 F1 on people subset, see Table 3), effectively serving as a strong upper-bound reference for the difficulty of evidence aggregation in our benchmark.

4.5 Ablation Study

We further investigate the impact of retrieval budget (top-k) on the performance of summary questions. We conduct an experiment using HippoRAG2 on a specific domain subset of our dataset, varying the top-k parameter from 2 to 12. As illustrated in Figure 5, the F1 score exhibits an inverted U-shaped trend: it initially increases as k grows, reaching a peak at $k = 8$, and subsequently declines as k increases further to 12.

This trend suggests that an optimal retrieval budget is crucial for balancing recall and precision. When top-k is too small (e.g., $k < 5$), the system fails to retrieve sufficient evidence chunks to

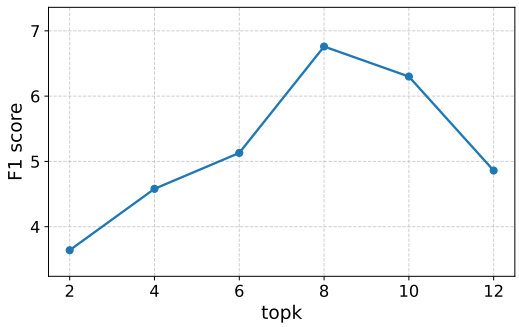

Figure 5: Impact of retrieval budget (top-k) on F1 score for summary questions. F1 increases as k grows, then drops when k is too large. It peaks at $k = 8$.

cover the broad factual content required for summarization, limiting recall. Conversely, when top-k becomes too large (e.g., $k > 8$), the introduction of excessive irrelevant noise or "distractor" chunks overwhelms the generator's context window or reasoning capability, leading to hallucinations or loss of focus, which degrades the overall F1. This finding emphasizes that retrieval strategies must be carefully tuned to the corpus size and noise level to maximize performance on complex summary tasks.

To verify that our conclusions generalize across model choices, we replace gpt-4o-mini with gpt-5-mini for both graph construction and QA in HippoRAG2 on the Culture domain. QA improves substantially by +17pp single-fact and +16pp multi-fact, while summary F1 barely changes at +1.65pp, confirming that retrieval, not generation, is the summary bottleneck. The task-difficulty ranking is consistent across both models, as shown in Table 7.

LLM	SF Acc.	MF Acc.	Sum. F1
gpt-4o-mini	62.79	24.32	6.68
gpt-5-mini	80.23	40.54	8.33

Table 7: LLM sensitivity analysis on HippoRAG2, Culture domain. Replacing gpt-4o-mini with gpt-5-mini improves QA but barely affects summary F1.

4.6 Error Analysis

We conduct a case study on the Culture domain with 37 multi-fact and 32 summary questions, using intermediate retrieval results to diagnose failure modes. Full details are in Appendix C. For multi-fact errors, 65% are generation-side, where the model omits or contradicts facts despite relevant retrieved documents, and 35% are retrieval-side. This suggests that even when retrieval succeeds, the LLM struggles to faithfully aggregate scattered facts from multiple wild-source documents. Summary errors decompose as 50% retrieval failure where gold information is absent from all retrieved documents, 43.7% generation failure including 12.5% hallucination, and 6.2% partial match. The high retrieval failure rate for summaries highlights the difficulty of achieving broad evidence coverage under noisy, heterogeneous corpora. Three wild-source-specific error patterns emerge: temporal confusion from conflicting dates across web pages, hub-entity shortcuts from high-degree graph nodes, and temporal over-representation from edge-density imbalance.

Question Type	Single-fact	Multi-fact	Summary
Agreement (%)	89.8	96.0	98.0

Table 8: Human–LLM judge agreement on 50 randomly sampled items per question type. All 8 disagreements are cases where the LLM judge was stricter than the human annotator.

4.7 LLM Judge Reliability

To quantify evaluation reliability, we randomly sample 50 items per question type from all scored outputs, stratified across methods and domains, and have a human annotator verify each LLM judgment. Table 8 reports the agreement rates. All 8 disagreements are cases where the LLM judge was too strict, a conservative bias desirable for benchmarking.

5 Conclusions

We introduce *WildGraphBench*, a benchmark designed to evaluate GraphRAG on wild-source corpora. Utilizing the heterogeneous reference pages cited in Wikipedia as the retrieval corpus while grounding statements in the Wikipedia articles themselves, we construct three progressively challenging task types to stress-test retrieval, aggregation, and summarization in uncurated environments. Our experiments reveal that while graph-based retrieval offers limited gains over strong flat baselines for simple, single-fact queries, it demonstrates significant advantages on multi-fact questions requiring cross-document evidence aggregation. Conversely, performance on summarization tasks remains low across all methods in this wild setting, highlighting the critical need for more robust evidence acquisition and synthesis mechanisms in real-world scenarios.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback, which helped improve the clarity and rigor of this work. We also thank the human annotators who participated in the evaluation and error analysis. This research is supported by Artificial Intelligence-National Science and Technology Major Project 2023ZD0121200, and Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under No.JYB2025XDXM103.

Limitations

Our benchmark derives gold statements from Wikipedia, which reflects editorial consensus rather than absolute truth; consequently, the gold set may inherit omissions or inaccuracies from Wikipedia and its citations. In addition, our evaluation relies on LLM-based judgment and statement matching, which may introduce systematic biases (e.g., preference for certain phrasing or verbosity) and may not perfectly mirror unbiased human assessment. These factors should be considered when interpreting absolute scores and fine-grained comparisons between methods.

Ethical considerations

Data and intended use. *WildGraphBench* is constructed from Wikipedia articles and their external reference pages; we use citation-linked Wikipedia statements as gold facts and the cited pages as a noisy retrieval corpus. We respect the original sources' licenses/terms and do not claim ownership over third-party content; any redistribution or derivatives should comply with the original access conditions. We specify *WildGraphBench*'s intended use as **research-only benchmarking** for retrieval robustness and multi-document evidence aggregation, and we discourage non-research uses unless explicitly permitted by the source conditions.

Risks. Because the corpus contains long, heterogeneous web pages, it may include noise, bias, outdated claims, toxic language, or inadvertently exposed sensitive information. Accordingly, we recommend standard safety practices (e.g., toxicity/PII filtering when appropriate) and emphasize that *WildGraphBench* evaluates robustness rather than establishing absolute truth.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

CircleMind-AI. 2024. Fastgraphrag: High-speed graph-based retrieval-augmented generation. https://github.com/circlemind-ai/fast-graphrag. GitHub repository.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *Preprint*, arXiv:2502.14802.

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.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *Preprint*, arXiv:2408.08921.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. *Preprint*, arXiv:2409.05591.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *Preprint*, arXiv:2104.07567.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *Preprint*, arXiv:2506.05690.

Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qian wen Zhang, Di Yin, Xing Sun, and Xiao Huang. 2025. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. *Preprint*, arXiv:2506.02404.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. *Preprint*, arXiv:2310.01558.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *Preprint*, arXiv:2501.13958.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. *Preprint*, arXiv:2510.10114.

A Dataset Statistics by Domain

Table 9 shows the per-domain question distribution in *WildGraphBench*.

Domain	SF	MF	Sum	Total
Culture	86	37	32	155
Geography	41	24	33	98
Health	76	19	55	150
History	25	1	10	36
Human act.	83	13	44	140
Mathematics	21	1	11	33
Nature	18	0	10	28
People	77	32	45	154
Philosophy	46	6	18	70
Religion	72	4	30	106
Society	66	21	27	114
Technology	56	33	24	113
Total	**667**	**191**	**339**	**1,197**

Table 9: Per-domain question counts in *WildGraphBench*. SF: single-fact; MF: multi-fact; Sum: summary.

B Results on every domain

Tables 10–21 report per-domain results for all evaluated methods. For each table, we include the following columns. **Avg** refers to the average accuracy across single-fact and multi-fact questions. **SF** refers to the accuracy on single-fact questions, where each question is grounded in exactly one gold statement. **MF** refers to the accuracy on multi-fact questions, which require aggregating evidence across multiple citations. **Rec**, **Prec**, and **F1** refer to the statement-level recall, precision, and F1 score for summary questions, respectively. All metrics are reported as percentages (%).

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	56.91	67.44	32.43	12.69	24.84	9.79
BM25	28.46	33.72	16.22	14.58	24.95	10.62
Fast-GraphRAG	20.32	22.09	16.22	5.73	24.64	2.48
HippoRAG2	51.22	62.79	24.32	11.02	19.79	6.68
MS GraphRAG(local)	41.46	43.02	37.84	3.39	12.96	1.61
MS GraphRAG(global)	52.04	52.33	51.35	6.77	15.54	1.80
LightRAG(hybrid)	46.34	52.33	32.43	14.25	21.36	6.93
LinearRAG	29.27	32.56	21.62	4.69	25.33	4.91

Table 10: Culture

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	60.00	70.73	41.67	10.92	22.83	7.89
BM25	35.38	43.90	20.83	6.84	14.62	4.81
Fast-GraphRAG	44.62	53.66	29.17	6.00	31.52	6.29
HippoRAG2	73.85	85.37	54.17	8.59	18.77	6.28
MS GraphRAG(local)	49.23	60.98	29.17	11.48	16.37	8.66
MS GraphRAG(global)	52.31	56.10	45.83	11.36	18.68	7.54
LightRAG(hybrid)	58.47	60.98	54.17	15.31	22.07	9.61
LinearRAG	60.00	65.85	50.00	5.71	38.66	5.87

Table 11: Geography

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	49.47	59.21	10.53	20.34	16.01	11.15
BM25	32.63	39.47	5.26	13.70	19.79	8.70
Fast-GraphRAG	31.58	36.84	10.53	10.79	17.23	6.87
HippoRAG2	62.10	71.05	26.32	16.59	14.23	9.12
MS GraphRAG(local)	31.58	34.21	21.05	15.60	10.53	7.01
MS GraphRAG(global)	54.74	60.53	31.58	20.60	12.41	10.15
LightRAG(hybrid)	52.63	59.21	26.32	17.73	14.23	9.90
LinearRAG	40.00	43.42	26.32	11.88	27.70	10.95

Table 12: Health

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	61.54	60.00	100.00	2.50	8.88	2.58
BM25	34.62	36.00	0.00	2.50	23.58	2.14
Fast-GraphRAG	34.62	32.00	100.00	0.00	23.35	0.00
HippoRAG2	73.08	72.00	100.00	3.33	8.06	2.07
MS GraphRAG(local)	46.15	44.00	100.00	0.00	17.12	0.00
MS GraphRAG(global)	65.38	64.00	100.00	5.00	12.53	5.45
LightRAG(hybrid)	69.23	68.00	100.00	0.00	12.49	0.00
LinearRAG	46.15	44.00	100.00	0.00	29.25	0.00

Table 13: History

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	81.25	85.54	53.85	12.60	24.20	8.09
BM25	50.00	51.81	38.46	10.38	20.32	6.50
Fast-GraphRAG	36.46	37.35	30.77	6.88	20.16	5.13
HippoRAG2	79.17	84.34	46.15	13.20	18.90	7.39
MS GraphRAG(local)	29.17	30.12	23.08	6.99	10.74	4.41
MS GraphRAG(global)	59.38	61.45	46.15	15.85	19.49	9.71
LightRAG(hybrid)	64.58	68.67	38.46	9.64	21.61	7.42
LinearRAG	51.04	50.60	53.85	4.26	19.97	3.71

Table 14: Human Activities

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	45.46	42.86	100.00	31.82	27.08	20.60
BM25	54.54	57.14	0.00	22.73	31.36	16.06
Fast-GraphRAG	50.00	47.62	100.00	36.36	29.13	24.97
HippoRAG2	54.54	57.14	0.00	27.27	32.63	20.22
MS GraphRAG(local)	45.46	47.62	0.00	31.82	23.46	18.23
MS GraphRAG(global)	54.54	57.14	0.00	39.39	15.48	19.20
LightRAG(hybrid)	59.09	61.90	0.00	40.91	20.31	22.32
LinearRAG	59.09	61.90	0.00	31.82	33.63	22.11

Table 15: Mathematics

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	50.00	50.00	0.00	13.33	23.86	6.50
BM25	38.89	38.89	0.00	8.33	11.87	5.06
Fast-GraphRAG	11.11	11.11	0.00	0.00	17.36	0.00
HippoRAG2	44.44	44.44	0.00	10.00	3.61	2.50
MS GraphRAG(local)	0.00	0.00	0.00	0.00	9.55	0.00
MS GraphRAG(global)	5.56	5.56	0.00	0.00	7.61	0.00
LightRAG(hybrid)	27.78	27.78	0.00	8.33	11.56	2.22
LinearRAG	22.22	22.22	0.00	0.00	44.01	0.00

Table 16: Nature

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	62.38	76.62	28.12	10.48	15.29	8.03
BM25	42.20	51.95	18.75	3.63	21.43	2.82
Fast-GraphRAG	29.36	33.77	18.75	1.48	22.83	1.62
HippoRAG2	62.39	72.73	37.50	7.63	15.69	6.14
MS GraphRAG(local)	33.95	38.96	21.88	4.59	9.17	2.98
MS GraphRAG(global)	55.05	62.34	37.50	5.52	14.13	5.41
LightRAG(hybrid)	58.71	67.53	37.50	5.56	15.69	4.73
LinearRAG	43.12	51.95	21.88	1.52	22.51	1.69

Table 17: People

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	46.15	50.00	16.67	7.41	8.69	2.44
BM25	26.92	28.26	16.67	1.39	6.17	0.79
Fast-GraphRAG	50.00	47.83	66.67	2.96	12.71	1.01
HippoRAG2	67.31	69.57	50.00	8.52	13.98	3.58
MS GraphRAG(local)	46.16	43.48	66.67	5.74	4.44	1.73
MS GraphRAG(global)	65.38	63.04	83.33	7.13	8.35	2.28
LightRAG(hybrid)	71.16	69.57	83.33	8.52	8.73	4.80
LinearRAG	50.00	50.00	50.00	1.85	17.79	2.56

Table 18: Philosophy

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	61.84	59.72	100.00	12.33	15.56	6.83
BM25	39.47	38.89	50.00	6.14	17.66	3.33
Fast-GraphRAG	34.21	30.56	100.00	3.81	20.06	2.50
HippoRAG2	67.11	68.06	50.00	8.87	19.88	5.94
MS GraphRAG(local)	27.63	27.78	25.00	3.45	10.12	1.74
MS GraphRAG(global)	50.00	47.22	100.00	8.73	13.37	2.65
LightRAG(hybrid)	60.52	58.33	100.00	9.48	19.95	5.47
LinearRAG	46.05	44.44	75.00	0.48	29.37	0.69

Table 19: Religion

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	62.07	74.24	23.81	17.31	19.22	12.89
BM25	28.73	33.33	14.29	8.64	18.00	5.58
Fast-GraphRAG	54.02	59.09	38.10	12.41	24.93	9.55
HippoRAG2	68.97	78.79	38.10	13.64	13.45	8.92
MS GraphRAG(local)	50.57	56.06	33.33	17.38	10.12	7.70
MS GraphRAG(global)	65.52	71.21	47.62	14.78	12.39	8.02
LightRAG(hybrid)	66.67	75.76	38.10	13.70	14.49	9.91
LinearRAG	50.57	56.06	33.33	8.77	28.99	10.14

Table 20: Society

Method	Avg	SF	MF	Rec	Prec	F1
NaiveRAG	57.30	64.29	45.45	10.87	24.03	11.91
BM25	40.45	44.64	33.33	6.11	22.38	5.45
Fast-GraphRAG	17.98	17.86	18.18	2.22	33.77	2.94
HippoRAG2	59.55	66.07	48.48	6.91	18.85	4.92
MS GraphRAG(local)	43.82	39.29	51.52	7.57	23.76	7.56
MS GraphRAG(global)	47.19	44.64	51.52	8.40	19.96	7.10
LightRAG(hybrid)	43.82	46.43	39.39	9.55	21.26	8.68
LinearRAG	47.19	48.21	45.45	2.22	36.39	3.00

Table 21: Technology

C Detailed Error Analysis

We provide detailed error analysis on the Culture domain subset with intermediate retrieval results for HippoRAG2. For each error pattern, we show a representative question, the gold answer, and the system output.

Temporal confusion. **Question:** *"When was the television series first aired, and in what year is it set?"*

Gold answer: First aired in 2025; set in the year 2027.

Prediction: "The series premiered in 2027." (incorrect: conflates air date with in-universe year)

The retrieved documents contain both "2025" (the real-world air date from a TV network press release) and "2027" (an in-universe year from a fan wiki page). Because these temporal references originate from different web pages with different contexts, the model conflates them and selects the wrong date.

Hub-entity shortcuts. **Question:** *"Which short-form digital series were produced under the Marvel Television banner?"*

Gold answer: WHIH Newsfront, Slingshot, and other short digital series.

Prediction: "The Defenders, Jessica Jones, Luke Cage..." (incorrect: retrieves full-length Netflix series instead of short digital ones)

The graph links "digital series" to the Netflix Defenders franchise via a high-degree "Marvel Television" hub node. This hub creates a shortcut path that bypasses the correct but low-degree entities (WHIH Newsfront, Slingshot), causing the system to retrieve extensive but irrelevant Defenders documentation.

Temporal over-representation. Question: *"Summarize the major milestones in Marvel Television's production history."*

Gold answer: Should cover events from 2013 through Phase Six announcements.

Prediction: Focuses heavily on the 2013 Netflix deal, with minimal mention of recent Phase Six developments. (incorrect: under-represents recent events)

Long-form articles from Variety and Deadline about the 2013 Netflix deal generate significantly more graph edges than short Phase Six press releases. This edge-density imbalance causes the graph traversal to systematically over-retrieve older events, under-representing recent announcements despite their relevance to the query.

D Prompts for Data Construction

Figure 6 shows the prompts used for data construction, including citation-aware statement extraction, question generation, and summary question filtering.

Question Generation: Single-fact

You are constructing a question for a **SINGLE-FACT (supported by one citation)** citation-based QA dataset.
ARTICLE TITL
{{WIKI_TITLE}}
SECTION PATH:
{{SECTION_PATH}}
WIKI SENTENCE (with inline citations):
{{SENTENCE}}
CLEAN FACTUAL STATEMENT (this will be used as the reference answer):
{{STATEMENT}}
REFERENCE URLS (for context, do NOT quote them explicitly):
{{REF_URLS_LIST}}
Your task:
- Write ONE natural-language QUESTION in English or Chinese (depending on the style of the article), such that:
  - The gold answer should be exactly the given STATEMENT (possibly with tiny paraphrasing).
  - The question should contain **multiple constraints** (e.g. entity + time, quantity + condition, entity + location).
  - If any of these constraints was removed, the question would become under-specified or wrong.
  - The question must be answerable solely from the given statement and sentence.
- The question should feel natural and non-trivial:
  - Do NOT copy any span of 4 or more consecutive words from the sentence or the statement.
  - Avoid generic patterns like "What is X?", "Who is Y?", "When did X happen?".
Return JSON ONLY:
{"question": "..."}

Question Generation: Multi-fact

You are constructing a question for a **MULTI-FACT (requires several citations together)** citation-based QA dataset.
ARTICLE TITLE:
{{WIKI_TITLE}}
SECTION PATH:
{{SECTION_PATH}}
WIKI SENTENCE (with inline citations):
{{SENTENCE}}
CLEAN FACTUAL STATEMENT (this will be used as the reference answer):
{{STATEMENT}}
REFERENCE URLS (for context, do NOT quote them explicitly):
{{REF_URLS_LIST}}
Your task:
- Write ONE natural-language QUESTION in English or Chinese (depending on the style of the article), such that:
  - The gold answer should be exactly the given STATEMENT (possibly with tiny paraphrasing).
  - The question should contain **multiple constraints** (e.g. entity + time, quantity + condition, entity + location).
  - If any of these constraints was removed, the question would become under-specified or wrong.
  - The question must be answerable solely from the given statement and sentence.
- The question should feel natural and non-trivial:
  - Do NOT copy any span of 4 or more consecutive words from the sentence or the statement.
  - Avoid generic patterns like "What is X?", "Who is Y?", "When did X happen?".
Return JSON ONLY:
{"question": "..."}

Figure 6: Prompts used for constructing questions (part 1).

Question Generation: Summary

```
You are constructing a TOPIC-CENTERED SUMMARY QUESTION for a topic.
TOPIC PATH (broad -> specific):
{{SECTION_PATH}}
OPTIONAL BODY EXCERPT (for natural phrasing only):
{{BODY_EXCERPT}}
GOLD STATEMENTS (facts that a good answer SHOULD cover; do NOT quote them):
{{GOLD_STATEMENTS_LIST}}
Your task:
- Write ONE natural-language question that asks for a concise, encyclopedic-style overview of the MOST SPECIFIC topic
  (typically the LAST 1–2 elements of the path).
- Use the GOLD STATEMENTS only as soft guidance to choose what aspects to emphasize,
  so that the answer naturally tends to cover those facts.
- The question must remain strongly anchored to the leaf topic in the path.
STRICT constraints:
- DO NOT mention Wikipedia/article/section/heading or similar meta words.
- DO NOT copy any span of 4+ consecutive words from any gold statement.
- Avoid leaking specific factual details from the gold statements in the question
  (especially exact numbers, exact dates, long proper names, or verbatim event descriptions).
  You may mention high-level aspects (e.g., "history", "structure", "major components", "development", "reception")
  if they align with the leaf topic and the gold statements.
- 20–200 characters.
Return JSON ONLY:
{"question": "..."}
```

Summary Question Filterring

```
You are doing a POST-HOC VERIFICATION for a citation-based summary dataset.
ARTICLE TITLE:{{WIKI_TITLE}}
LEAF SECTION TOPIC PATH:{{SECTION_PATH}}
Leaf topic (most specific):{{LEAF_TOPIC}}
You will be given multiple ITEMS. Each item has:
- a STATEMENT (candidate gold statement)
- several REFERENCES (content excerpts)
Task:
For EACH item:
- keep=true only if the REFERENCES (collectively) contain enough information to support ALL key factual claims in the STATEMENT.
- keep=false if key facts are missing, contradicted, or the references are irrelevant/noisy.
Rules:
- Use ONLY the given references; ignore outside knowledge.
- Be fairly strict: if unsure due to missing evidence, set keep=false.
# [Items are dynamically inserted here in the following format:]
# ### ITEM 1
# STATEMENT: ...
# REFERENCES: #### REF 1.1 ...
# ...
Return JSON ONLY:
{"items":[{"idx":1,"keep":true/false,"reason":"brief"}], "summary":"brief"}
```

Figure 6: Prompts used for constructing questions (part 2).

Single-fact Question Filtering

```
You are checking whether the provided REFERENCES collectively support a Q&A.
Q: {{QUESTION}}
A: {{ANSWER}}
REFERENCES (may include some noise, read holistically):
{{REFERENCES_CONTENT_BUNDLE}}
Rules:
- If the references together contain the key facts to justify the answer, return supported=true.
- If key facts are missing or contradicted, return supported=false.
Return JSON ONLY (do NOT explain your reasoning process, be concise):
{"supported": true/false, "reason": "brief"}
```

Multi-fact Question Filtering

```
You are given a factual STATEMENT (used as the reference answer for a QA pair),
together with the QUESTION and several reference documents cited from Wikipedia.
QUESTION:
{{QUESTION}}
REFERENCE ANSWER (STATEMENT):
{{STATEMENT}}
REFERENCES:
{{REFERENCES_CONTENT_BUNDLE}}
Your task is to judge whether these references are **jointly necessary**
to support the FULL factual content of the STATEMENT.
Rules:
1. Consider ONLY the information contained in the given references. Ignore any outside world knowledge.
2. For EACH reference individually, imagine you only had that single reference:
   - If that single reference ALONE already contains enough information to support
     ALL key factual claims in the STATEMENT (numbers, named entities, relationships,
     important conditions), then that reference is "individually sufficient"
     to justify the STATEMENT.
3. If **ANY** single reference is individually sufficient, then the multi-reference pattern is
   NOT truly necessary.
   → In this case, set all_needed = false.
4. Only if **NO** single reference is individually sufficient (each one misses some essential facts),
   and you really need to COMBINE at least two references to cover the full STATEMENT,
   set all_needed = true.
"Key factual claims" means the main facts expressed by the STATEMENT,
not minor stylistic details.
Return JSON ONLY in the following format:
{"all_needed": true/false, "reason": "brief"}
```

Figure 6: Prompts used for constructing questions (part 3).