

Peer Review

Review of: "SiReRAG: Indexing Similar and Related Information for Multihop Reasoning"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper argues that RAG indexing has been built on one organising principle at a time, and that this is why multihop questions go wrong. RAPTOR groups text by semantic similarity and summarises recursively; HippoRAG and GraphRAG group text by entity structure. The authors call the first *similarity* and the second *relatedness* (§1, definitions given in the third paragraph), and propose SiReRAG, which builds both trees and flattens every node from both into one retrieval pool (§4.3). The similarity side is RAPTOR unchanged, four levels deep. The relatedness side is new: a fine-tuned Mistral-7B-Instruct-v0.3 extracts propositions and their entities from each chunk (§4.2), propositions sharing an entity are concatenated by exact string match into *proposition aggregates*, and RAPTOR-style recursive summarisation runs on top of those aggregates as if they were pseudo-documents.

The empirical claims are: consistent gains over RAPTOR, HippoRAG and GraphRAG on MuSiQue, 2WikiMultiHopQA and HotpotQA, 1,000 validation questions each, with "an average 1.9% improvement in F1 scores" (abstract) and "up to 7.8% improvement in average F1 scores" when SiReRAG's pool is handed to BM25 or ColBERTv2 (abstract, §6.3); an ablation showing that proposition aggregates carry most of the benefit (§6.2); and an efficiency argument that the roughly threefold larger pool does not cost proportionally more time (§6.4, Table 7, Appendix C). Two things deserve credit before I start objecting. The first is §4.1: the authors tested an abstraction-level tree design on QuALITY, got 78.88 for RAPTOR against 78.76 for their hierarchical variant (Table 2, Average Accuracy column), and reported the negative result plainly rather than burying it. The second is that §3 attempts to motivate the architecture with a measurement instead of an assertion. Both are more than most papers in this area do.

I should state a limitation of my own review up front. As rendered on Qeios, Tables 4, 5, 6, 8, 9 and 11 are not legible: the scraped page carries their captions but not their cells. Every headline accuracy number in this paper lives in Table 4, and I could not check a single one. What follows is therefore an assessment of the design, the reasoning, and the tables that did survive (1, 2, 3, 7 and 10, plus Appendix C).

2. Whether the conclusion holds on the data

2.1 The motivating experiment does not test the proposed mechanism. This is my main objection. Section 3 operationalises relatedness as GPT-4o topic extraction followed by GMM clustering over topic embeddings ("we first extract the topic of each passage using OpenAI GPT-4o and then cluster passages based on the representations of their topics"). Section 4.2 operationalises relatedness as shared entities matched by exact string. Those are different constructions, and they can disagree badly: two passages about the same entity often have different topics, and two passages on the same topic frequently share no entity string at all. Table 1 measures the first, and the paper spends the rest of its length building the second. The bottleneck the authors demonstrate is therefore not the bottleneck their method addresses. The fix is small in principle: rerun the Table 1 coverage analysis with the entity-shared-proposition grouping actually used in §4.2.

2.2 Table 1 is internally coherent, and its own numbers undercut the framing. I checked the consistency of the overlap columns. If the overlap ratios are 50.15% (Overlap@Similarity) and 68.85% (Overlap@Relatedness) over the same intersection, then the similarity and relatedness sets stand in the ratio $68.85 / 50.15 = 1.373$; the coverage columns give $19.14 / 13.94 = 1.373$. Exact agreement, which tells me these cells were computed rather than estimated. Good. But the union follows from the same three numbers, and the paper never computes it: $|S \cup R| = 19.14 + 13.94 - (0.5015 \times 19.14) = 33.08 - 9.60 = 23.48\%$ of gold pairwise connections. So the ceiling on what combining both philosophies can reach, measured on the authors' own MuSiQue setup, is 23.5% against 19.14% for similarity alone. That is 4.3 points of headroom, not a doubling. I think this number belongs in §3 explicitly. It is honest, it still supports the paper's direction, and it sets the reader's expectation for a 1.9-point F1 gain instead of leaving the gap between motivation and result unexplained.

2.3 The comparison is not context-matched, and §5.4 says so in passing. All methods retrieve the top 20 candidates. SiReRAG's pool is far larger (Appendix C: 35,070 vs 12,371 on MuSiQue, 19,100 vs 6,939 on 2Wiki, 29,934 vs 10,031 on HotpotQA, i.e., ratios of 2.84, 2.75, and 2.98) and its extra members are proposition aggregates and their summaries, which are by construction denser and longer than a raw

chunk: Table 3 reports a maximum of 168, 195, and 165 propositions per entity on the three datasets. Twenty SiReRAG nodes are therefore not twenty RAPTOR nodes. They are plausibly several times the token count. The paper states the underlying reasoning outright in §5.4: "SiReRAG is expected to perform better when retrieving more due to the incorporation of proposition aggregates and their recursive summaries." That sentence concedes that $k = 20$ was chosen in a way that suits the method. Nowhere in the paper did I find the number of tokens delivered to GPT-4o per query for each system, so a reader cannot tell whether the gain comes from better organisation or simply from more evidence in the prompt. The experiment that settles it is cheap: match retrieved context by token budget rather than by k , and sweep k for both systems.

2.4 No variance, no test, on any main result. The authors know their pipeline is stochastic. The caption of Table 2 says it: "Due to randomization of clustering and generation temperature of LLMs, we run each indexing method 5 times." They then ran the main comparison, as far as the text reveals, once. Tables 4, 5, and 6 carry no standard deviations and no confidence intervals, and I found no significance test anywhere in the paper. The word "significantly" is used at least three times about differences that were never tested: §6.1 ("SiReRAG achieves significantly higher performance than indexing baselines"), §6.2 ("their removal resulting in a significant performance decline"), §6.3 ("having SiReRAG benefits both BM25 and ColBERTv2 significantly"). With $n = 1,000$ questions and paired outputs, a paired bootstrap or a permutation test over per-question F1 costs minutes and would settle the 1.9-point headline. Until it is run, the claim of consistency rests on point estimates from single runs of a randomised indexing procedure whose own five-run standard deviation, in the one place it was measured (Table 2), is of the same order as the effect being claimed.

2.5 Table 3 does not reconcile. The column is headed "#Props per Entity" with an Avg. sub-column. Divide propositions by entities and you get $54,605 / 50,926 = 1.07$ on MuSiQue, nowhere near the stated 2.74. Divide propositions by proposition aggregates instead: $27,697 / 11,108 = 2.493$ on 2Wiki, which matches the stated 2.49 exactly. So the column is really propositions per aggregate. But then the other two rows fail: MuSiQue gives $54,605 / 20,788 = 2.63$ against a stated 2.74, and HotpotQA gives $47,153 / 18,278 = 2.58$ against a stated 2.66. One of the three rows reconciles and two do not, under either reading. Something in that table is wrong, and since it is the only quantitative description of the relatedness tree, it matters.

2.6 The entity counts suggest the grouping barely groups. Table 3 puts entities and propositions at near parity on all three datasets: 50,926 vs 54,605 on MuSiQue, 46,856 vs 47,153 on HotpotQA, and on 2Wiki

there are actually *more* entities (29,490) than propositions (27,697). Since aggregation is by exact string match (§4.2, "using exact match"), any surface variation produces a separate entity: the paper's own Appendix D example contains both "Sir Nicholas Bacon" and "Nicholas Bacon". Near-parity counts are what you would expect if a large share of extracted entity strings occur exactly once and form aggregates of size one, which are just propositions under another name. The paper needs a distribution, not an average: what fraction of entities yields an aggregate of size 1, and what does performance look like when those are dropped? Note that this connects to the ablation: variant (B), which adds individual propositions to the pool, hurt performance, and the authors read that as redundancy. An alternative reading is that singleton aggregates already put those propositions in the pool, so (B) mostly duplicates them.

2.7 The efficiency metric cannot fail. TPER (Eq. 1) is the ratio of inference times divided by the ratio of pool sizes, and §5.3 declares that a value below 1 shows reasonable efficiency. I verified all four TPER cells in Table 7 and they are arithmetically correct: MuSiQue, $(2.653 / 1.560) / (35,070 / 12,371) = 1.701 / 2.835 = 0.600$; 2Wiki, $(1.974 / 1.437) / (19,100 / 6,939) = 1.374 / 2.753 = 0.499$; HotpotQA, $(2.319 / 1.502) / (29,934 / 10,031) = 1.544 / 2.984 = 0.517$; and the Average column is the mean of those three, 0.539. The problem is what the metric measures. TPQ includes the GPT-4o answer generation, which is the same work in both systems. Fitting a constant-plus-linear model to the MuSiQue pair gives a marginal cost of $(2.653 - 1.560) / (35,070 - 12,371) = 4.8 \times 10^{-5}$ seconds per node, so RAPTOR's own retrieval accounts for about 0.60 s of its 1.560 s and generation accounts for roughly a second. With a fixed cost of that size in both numerator terms, TPER falls below 1 automatically unless retrieval time grows superlinearly in pool size, which for a flat embedding search it never will. The metric is therefore a restatement of the fact that generation dominates, not evidence about the index. Report retrieval latency separately from generation latency and the claim becomes checkable.

2.8 Indexing cost is absent, and it is the cost that matters here. Building the relatedness tree takes a Llama-3-70B pass over 10K IndustryCorpus documents to make training data, a fine-tuning run on Mistral-7B, an extraction pass over every chunk of every evaluation corpus (11,656 chunks on MuSiQue alone, Table 3), and recursive summarisation over 20,788 aggregates. None of this appears in any table. §6.4 and Appendix C discuss only query-time cost, where the method looks cheap. A paper whose abstract calls the method "a reasonably efficient solution" should report GPU hours, API token counts, or wall-clock time for index construction against RAPTOR's.

2.9 Two smaller characterisation issues. First, §6.4 calls SiReRAG's overhead "slightly longer inference time", but Table 7's Average TPQ column is 2.315 against RAPTOR's 1.500, which is 54% more time per

query. Second, Appendix B states that adding SiReRAG raises self-ask TPQ "by approximately 1.2 seconds over the three datasets". That is the average (Table 10, Average TPQ: $3.63 - 2.41 = 1.22$), but the per-dataset increases are 1.81 s on MuSiQue, 0.86 s on 2Wiki, and 1.01 s on HotpotQA. On MuSiQue, that is a 67% increase, and the phrasing hides it.

2.10 The GraphRAG comparison inflates the apparent margin. The authors are candid that GraphRAG targets query-focused summarisation, that they had to use local search rather than global, and that "GraphRAG often provides 'I don't know' answers" (§6.1). Under EM and F1, an abstention scores zero. Reporting "more than 20% higher than GraphRAG on average EM and F1" is then close to measuring the abstention rate. Either report the abstention rate alongside, and the scores on the answered subset, or drop the baseline and say why.

3. Reproducibility

Some of this is handled well. The open-weight models are pinned with links to exact HuggingFace revisions in the footnotes: Meta-Llama-3-70B-Instruct and Mistral-7B-Instruct-v0.3. The embedding model is named (text-embedding-3-small). The corpora are the HippoRAG corpora, which is a real anchor. All the LLM prompts are given in Appendix F and Figures 5 to 7, including the summarisation prompt and the answer prompt ("answer this question in as fewer number of words as possible"). The proposition definition is quoted verbatim in §4.2. That is better than the median for this literature, and I want to say so.

The gaps are nonetheless serious, and several are the ones that come up in every LLM paper.

No snapshot dates for the closed models. §5.4 says GPT-4o and GPT-3.5-Turbo without a dated identifier such as gpt-4o-2024-08-06. Both aliases moved repeatedly during 2024. Since GPT-4o performs the answer generation for every system in Table 4, and also the topic extraction in §3, the entire result set is pinned to an alias that no longer resolves to the same weights.

No decoding parameters. Temperature, top-p, and max tokens are never given for any call: not for summarisation, not for extraction, not for answer generation. The Table 2 caption refers to "generation temperature of LLMs" as a source of run-to-run variance, which implies it is above zero, but the value is not stated anywhere I could find.

No seeds, and no code. I searched the text for a repository, a data release, or an availability statement and found none. This matters more than usual because the relatedness tree depends on a fine-tuned Mistral-7B that exists only on the authors' machines. Without the checkpoint or the extraction outputs, nobody

can rebuild the relatedness tree, which means nobody can reproduce the paper's central claim. Releasing the extracted propositions and entities for the three corpora would go most of the way even if the checkpoint stays private.

Fine-tuning is described in one sentence. §4.2 says training data was consolidated from 10K BAAI/IndustryCorpus documents and used to fine-tune Mistral-7B-Instruct-v0.3. No count of training examples, no learning rate, no epoch count, no batch size, no statement of whether this was full fine-tuning or an adapter, no held-out evaluation of extraction quality. There is also no evidence at all about whether the extracted propositions are faithful to their source chunks, and the whole relatedness tree is built on them.

Clustering configuration. GMM soft clustering is inherited from RAPTOR, but its settings are not restated: component count or selection criterion, covariance type, the soft-assignment probability threshold, dimensionality reduction if any. The similarity tree is stated to have 4 levels (§1); the relatedness tree's depth is never given.

Question selection. §5.1 says "we obtain 1000 questions from each validation set", with no sampling rule. §3 separately draws 1,000 MuSiQue validation questions. Whether those are the same 1,000 is not stated, and if they are, the motivating analysis and the evaluation share data.

Hardware and cost. No GPU type, no count, no runtime. TPQ figures in Table 7 are in seconds, but the machine that produced them is not described, so they are not comparable to anything outside this paper.

4. *What specifically to fix*

1. **§3 and Table 1.** Rerun the coverage analysis using the actual relatedness construction of §4.2 (propositions grouped by shared entity via exact match), not GPT-4o topic clustering. Report it beside the current topic-based row so the reader can see both. While you are there, add the union: $19.14 + 13.94 - (0.5015 \times 19.14) = 23.48\%$, which is the ceiling your architecture is aiming at.
2. **§5.4, retrieval depth.** Top-20 for a 35,070-node pool of aggregates and summaries is not top-20 for a 12,371-node pool of chunks. Add a context-length-matched comparison: fix the token budget delivered to GPT-4o and let k vary per system. Report the mean retrieved-context token count per query for every row of Table 4.
3. **Tables 4, 5, 6 and 11: statistics.** Run the full pipeline at least three times per configuration and report standard deviations, as you already did for Table 2 on QuALITY. Add a paired significance test over per-question F1 (bootstrap or permutation) for SiReRAG against RAPTOR and against

HippoRAG on each dataset. Then either keep the word "significantly" in §6.1, §6.2 and §6.3, or remove it.

4. **Table 3.** The "#Props per Entity" column does not reproduce from the other columns. 2Wiki works as propositions over aggregates ($27,697 / 11,108 = 2.49$); MuSiQue and HotpotQA do not (2.63 vs 2.74 stated; 2.58 vs 2.66 stated). Fix the numbers or fix the header, and say which quantity is being averaged.
5. Add a histogram of aggregate sizes per dataset, and state what share of aggregates contain exactly one proposition. Given that 2Wiki reports more entities (29,490) than propositions (27,697), I suspect that share is large.
6. **§4.2, entity matching.** Say what normalisation, if any, precedes exact match, and report how much performance changes under a lenient variant (lowercasing, title stripping, or an alias table). Your own Appendix D example contains "Sir Nicholas Bacon" and "Nicholas Bacon".
7. **§5.3 and §6.4.** TPER as defined cannot exceed 1 while a fixed generation cost dominates TPQ. Split Table 7 into retrieval latency and generation latency, and compute TPER on retrieval alone. If you keep the current definition, state in §5.3 what value would have counted as a failure.
8. **New table: indexing cost.** Wall-clock, GPU hours and token counts for building each index (RAPTOR vs SiReRAG) on each of the three corpora, including the Llama-3-70B data-generation pass and the Mistral-7B fine-tune. The abstract's efficiency claim is not supported by query-time numbers alone.
9. **§6.1, the GraphRAG row.** Report the abstention rate you observed manually, and F1 restricted to answered questions.
10. **§5.4 and §4.2, versions.** Replace "GPT-4o" and "GPT-3.5-Turbo" with dated snapshot identifiers, and state temperature and max tokens for each call type. Add the fine-tuning hyperparameters and the training set size.
11. Release the extracted propositions and entities for MuSiQue, 2Wiki and HotpotQA, plus the indexing and evaluation code. Without them the relatedness tree cannot be rebuilt by anyone.
12. **§5.1.** State how the 1,000 questions per dataset were chosen and whether §3 uses the same 1,000 MuSiQue questions as §6.
13. **Abstract and §6.1, units.** The abstract says "an average 1.9% improvement in F1 scores" while §6.1 says "approximately 5% higher than RAPTOR on average F1". Say explicitly whether these are absolute points or relative percentages, and against which baseline each figure is computed.
14. **Appendix B.** Replace "increases by approximately 1.2 seconds over the three datasets" with the per-dataset values from Table 10: 1.81 s, 0.86 s and 1.01 s. Similarly, §6.4's "slightly longer inference time"

describes a 54% increase (Table 7, Average TPQ 2.315 vs 1.500) and should be stated as such.

15. Typos, all quick: "the frst supervised approach" (§2), "indices an entire corpus" should be "indexes" (§1), "This interaction may enables" (§4.3), "MiSuQue" (§6.3), "benifits" (Table 9 caption), and several missing spaces after the model name: "SiReRAGoutperforms" (§6.1), "SiReRAGfor indexing" (§6.1), "SiReRAGdemonstrates" (§6.4), "SiReRAG(sec 6.2)" (§6, RQ 3 and RQ 4).

5. Recommendation

Major revisions.

The idea is sound, and the paper is carefully written. Indexing by two complementary principles and pooling the nodes is a natural step past RAPTOR, the proposition-aggregate construction is a real contribution rather than a relabelling, the qualitative walkthrough in Appendix D is genuinely illuminating about why the extra node helps, and Appendix E extends the evaluation to single-hop, MultiHop-RAG, and ASQA instead of stopping at the three datasets that flatter the method. Reporting the failed hierarchical-tree experiment in §4.1 was the right call, and I hope the authors keep it.

My reason for major rather than minor revisions is that three of the paper's load-bearing claims are not currently supported by what is reported. The gain over baselines is a single-run point estimate from a randomised pipeline, with no variance and no test, while the method retrieves 20 items from a pool nearly three times larger and made of denser nodes, so the retrieval comparison is not matched on the quantity that most plausibly drives the result. The motivating analysis in §3 measures a different notion of relatedness than the method implements. And the efficiency claim rests on a ratio that cannot come out badly, computed on query-time only, while the index construction that the method adds is never costed. None of these requires new machinery, and two of them (paired significance tests, token-matched retrieval) are a few days of compute on experiments the authors have already built.

Table 3 should also be corrected before publication; I could not make two of its three rows reconcile under any reading of the column header. And I would ask for the extracted propositions to be released. Otherwise, the central artifact of the paper sits behind a fine-tuned checkpoint that nobody else has.

Declarations

Potential competing interests: No potential competing interests to declare.