

RIPPLEKB: FINDING WHAT AN EDIT CHANGES ACROSS LINKED DOCUMENTS

Anonymous authors

Paper under double-blind review

ABSTRACT

Editing one fact can change a derived quantity, comparison, or other statement across linked documents. RippleKB evaluates whether systems can recover a constructed set of affected source units completely. Each item edits one source fact in a small set of linked documents. The proposed answer key is a hidden dependency closure, with unaffected units also present. A system returns a ranked list of source units, word for word, under a review rule tied to the constructed affected-set size. We regenerate every item from a public multi-hop corpus and test identifier-order, document-position, lexical, and supporting-fact rankings, alongside role and template concentration checks. We release the audit protocol under which the proposed closure is to be reviewed unit by unit. We report how far retrieval-style and language-model baselines get toward recovering the constructed closure completely on the machine-validated candidate set. Complete recovery tests whether a ranking includes the entire constructed target, while exact-copy measures separately assess preservation of source text.

1 INTRODUCTION

An edit can be locally correct while leaving a collection of documents inconsistent. A changed quantity can alter a total in another document and an explanation farther along the chain. Finding the edited sentence alone leaves those consequences unreviewed. Change-impact analysis studies dependencies in requirements and software artifacts; document-update research studies revisions prompted by new evidence (Goknil et al., 2016; Etezadi et al., 2025; Logan IV et al., 2022).

The difficult endpoint is finding every affected statement. A system can retrieve useful evidence or produce a correct revised answer while missing statements outside that answer’s evidence path. Document editing, knowledge editing, and multi-hop question answering evaluate related capabilities through revised text, changed model knowledge, or evidence for a chosen question (Kruthof, 2026; Dwivedi-Yu et al., 2024; Cohen et al., 2024; Zhong et al., 2023; Yang et al., 2018). RippleKB makes the affected source units themselves the target.

The target is an **impact set**: the source units whose truth or content changes because of the edit. RippleKB proposes it through a **dependency closure**, the set reached through an item’s hidden proposed dependencies. The system ranks each **verbatim unit**, a source unit reproduced exactly as it appears in its document. Unaffected statements remain among the candidates, so the ranking determines where the complete constructed target appears. Figure 1 follows a conceptual stock-count edit through a changed total and comparison, then shows an affected unit left beyond the ranking’s review cutoff.

Partial recovery and complete recovery therefore answer different maintenance questions. One measures the fraction of the impact set found; the other asks whether any affected unit remains missing. The scorer applies the same size-based budget rule to each ranking, using an affected-set size hidden from the system. Selection and copying are also distinct: a correct identifier can retrieve a unit even when its submitted text is altered.

The construction makes the proposed answer key explicit and tests specified structural and lexical routes to recovering it. We evaluate retrieval and language-model rankings on the resulting machine-validated candidates. Embedding reranking raises BM25’s partial-recall point estimate from 0.66 to 0.68 while lowering complete recovery from 20% to 13%. A later-added language model reaches

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

Assume $n + m < q \leq n + \delta + m$ so the comparison changes.



A North stores n crates.
 B Combined stock is $n + m$ crates.
 ... other unchanged units ...

scorer's top K cutoff

C Combined stock is below q crates.

A and B are copied exactly.
C falls below the cutoff.
Full coverage needs A, B, C;
returned words stay unchanged.

Conceptual example; lists are abbreviated and the ranking is illustrative.

078
079
080
081
082
083
084
085
086
087

081
082
083
084
085
086
087

084
085
086
087088
089

090
091
092
093
094
095

096
097
098
099
100
101
102

103
104
105
106
107

that condition explicitly. The dependency is therefore about the consequence of this edit under the stated rule, rather than the mere occurrence of a related quantity.

The unchanged statements in the example make this distinction visible. A statement about when the same store opens shares its subject with the edit but is unaffected by the count change. The other store’s count contributes to the total while remaining unchanged itself. The target follows the edit’s consequences, so topical similarity and participation in the surrounding calculation are different reasons for a unit to appear in the documents. The hidden graph records the proposed affected set; the visible text supplies what the system can use to recover it.

Figure 1 illustrates the output contract; Section 3 gives the numeric-edit and lexical constraints used for actual candidates. Returning a verbatim unit means copying its supplied text, not rewriting it to incorporate the new value. This keeps finding affected material separate from deciding how to revise it. The output contract asks the system to rank every candidate unit; the scorer also accepts shorter lists, with missing positions receiving no credit. Each entry contains its identifier and source text. An unknown or duplicate identifier consumes a position without adding selection credit. A valid affected identifier can receive selection credit even when its text is copied incorrectly; fidelity scores measure that separate error.

The review budget is an item-specific scoring allowance, determined by a common rule. For candidate set U and target set G from the constructed dependency closure, the larger budget is $K = \min(|U|, 2|G|)$. The scorer knows $|G|$; the system ranks candidates without seeing that size or its resulting cutoff. Returning every candidate therefore earns credit according to its order within the same budget. Appendix E gives the duplicate, malformed-output, and missing-position rules.

Partial recovery and completed review sets. Three retrieval measures describe the same ranking at complementary resolutions. Recall@2G is the mean fraction of affected identifiers recovered within K positions; complete-closure@2G is the proportion of items whose entire impact set is recovered there. Recall@G uses the smaller budget $\min(|U|, |G|)$. The recall average gives each item equal weight, regardless of the size of its impact set. Complete recovery instead records an all-or-nothing outcome for each item. Writing $P_i(k)$ for the distinct valid identifiers recovered in the first k positions makes the two per-item scores explicit:

$$r_i = \frac{|P_i(K_i) \cap G_i|}{|G_i|}, \quad c_i = \mathbf{1}\{G_i \subseteq P_i(K_i)\}, \quad K_i = \min(|U_i|, 2|G_i|).$$

The reported recall and completion scores average r_i and c_i across items. The stricter prefix asks which units the ranking prioritizes; the larger prefix gives room for unaffected units while still requiring all affected identifiers for completion.

The larger budget gives random selection substantial partial credit on the current candidate collection. The retained random baseline reaches Recall@2G 0.54 alongside complete-closure@2G 1%, with Recall@G 0.27. These are the pooled baseline scores used in Section 5.

Finding a unit and preserving its bytes. Canonical comparison converts transport CRLF line endings to LF and otherwise requires exact source bytes. Numeric and named-entity fidelity likewise count preserved annotated spans, with missed affected units contributing no recovered spans. Conditional verbatim exactness considers only recovered affected identifiers and pools those records across items. Numeric and entity scores pool their respective annotated spans rather than averaging item-level fractions. Each span must retain its bytes at the recorded position in the returned unit; reformatting a number or shifting a span can fail this test without changing its meaning. Appendix E gives the remaining ranking and fidelity definitions.

Table 1 collects the questions answered by these measurements.

3 CONSTRUCTION AND ANSWER-KEY REVIEW

Construction must supply a target that is explicit enough to score and whose proposed dependencies can be assessed against the documents. Mechanical consistency, shortcut controls, and semantic review address different parts of that requirement. The collection contains 150 machine-validated candidate items.

Table 1: What each measurement asks about the returned ranking. Set recovery, complete review, and source preservation retain their different denominators.

Measurement	Question and aggregation
Recall@2G	How much of each target was found? Mean of item fractions.
Complete-closure@2G	Was the whole target found? Fraction of completed items.
Conditional exactness	Was recovered text copied exactly? Pools recovered affected units.
Span fidelity	Were annotated bytes preserved? Pools all affected numeric or entity spans.

3.1 SOURCE ASSIGNMENT AND CANDIDATE GENERATION

Each item is regenerated from MuSiQue, whose questions compose facts across Wikipedia passages (Trivedi et al., 2022). Public source identifiers are assigned to development, selection, and held-out report partitions before construction. Source records used by another in-review work on the same corpus were excluded before assignment. Appendix K records source isolation.

The generation process fixes the edit, dependency shape, and document placement before producing downstream text. The edit changes a numeric value in a source-grounded sentence. A generator supplies proposed dependent statements and the relation text needed to connect them. Eight dependency shapes vary the impact-set size and depth, while unaffected units provide alternatives within every document. Each proposed dependency closure crosses a document boundary and contains mostly units outside the source question’s supporting facts.

Graph reachability proposes which units change; the documents must still support that interpretation. A statement that eligibility depends on a count, for example, needs a rule connecting the particular count change to a different eligibility outcome. The edge records the intended dependence; the released review protocol specifies how to assess it against the rendered statements.

Mechanical checks enforce the construction’s explicit contract. They verify required units and edges, source grounding, placement, unique unit text, and the alignment of annotated spans. Affected downstream units begin with their annotated entity, while unaffected units use unconstrained sentence openings. The checks also prevent tokens from the edited values from appearing in titles and downstream units, limiting a direct lexical cue. Public identifiers are assigned independently of construction roles, and document and unit orders are shuffled. The first mechanically valid proposal in each predetermined source reserve is selected (Appendix B).

3.2 TESTING ROUTES TO THE ANSWER KEY

Identifier ordering tests both sort directions; document-position ranking learns affected positions from development items and applies that ordering to the other partitions. Lexical bridging ranks units by BM25 or token overlap with the edit. The supporting-fact control places the original question’s support units first, followed by a seeded ordering of the remaining units. It uses hidden support labels as a privileged diagnostic of how far that evidence alone reaches. Evidence sufficient for that question may cover only part of the impact set.

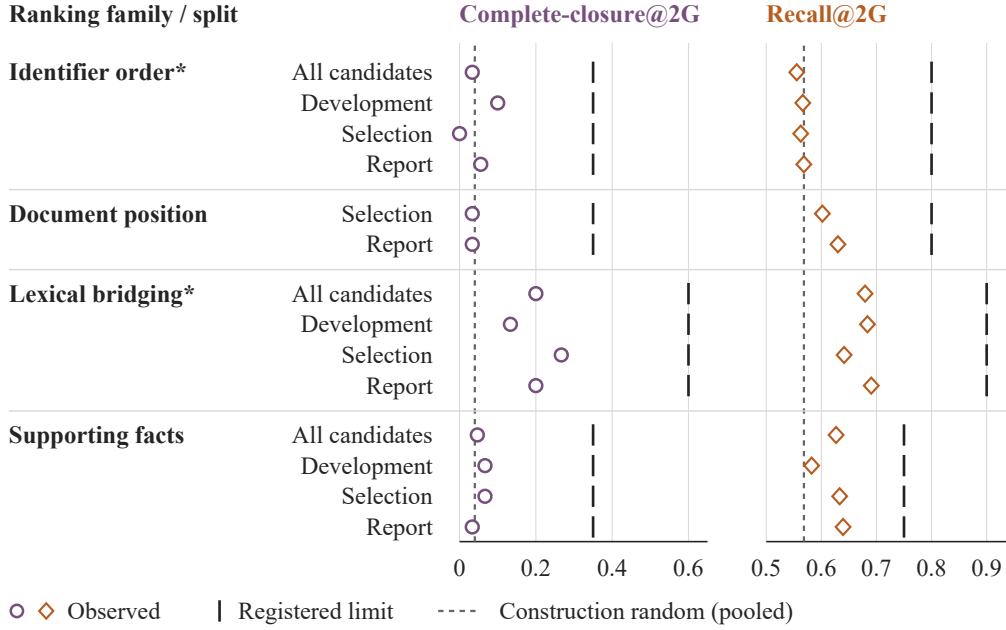
Role and template concentration is a composition check rather than a ranking method. It examines dependency-shape coverage, affected-position patterns, diversity, and uniqueness. Figure 2 separates these composition measurements from rankings; Appendix C gives the five checks’ thresholds and full results.

Lexical bridging is the strongest tested shortcut, completing at most 20% of pooled candidates and 27% of the selection split. All five checks meet their registered criteria; the entity-first openings of affected downstream units remain a potential cue outside these rankings. The plotted lexical and identifier summaries take separate maxima for each metric. Appendix C gives split coverage, thresholds, and the distinct random orderings used for construction diagnostics and baseline evaluation.

Can simple structure identify the affected units?

Construction checks against the constructed dependency closure

Lexical selection: complete 26.7% (BM25); recall 0.64.



* Per-metric maxima can come from different strategies.

Composition: max. shape 15.3%; affected-position pattern 1.3% (not retrieval rates).

Figure 2: Can the specified shortcuts recover the whole closure? Ranking controls pair complete recovery with recall; the separate composition checks measure concentration of dependency shapes and affected positions.

3.3 CONSTRUCTED TARGETS AND TEXTUAL SUPPORT

The reported scores measure recovery of the construction-defined closure on machine-validated candidates. The released review protocol assesses whether the visible text supports each proposed dependency and whether other affected units are missing from the closure (Appendix D).

3.4 CANDIDATE COMPOSITION

The construction assigns 30 candidates to development, 30 to selection, and 90 to the report partition. The present evaluation pools all 150 candidates without tuning on any split; Appendix G retains the available partition-level scores. Appendix B gives the dependency-shape, document-count, impact-size, and depth composition.

4 EVALUATION PROTOCOL

The comparison measures how early systems rank a complete impact set on the same 150 machine-validated candidates. The initial configurations were fixed before candidate scoring, and no setting was tuned on the development, selection, or report partition. The pooled evaluation uses all construction partitions, whereas the original plan specified a report-only population. Appendix A gives the source-cluster uncertainty procedure and Appendix F records the retained runs.

The initial comparison spans deterministic rankings, retrieval with embedding reranking, and language models. Deterministic methods use random, rendered, or identifier order, token overlap, or BM25

relevance to the edit. The retrieval baseline reranks BM25 candidates using Titan Text Embeddings V2. Qwen3 32B, Llama 3.3 70B, and Claude Sonnet 4.5 receive the edit and documents and rank candidate identifiers with each verbatim unit’s original text.

Claude Sonnet 5, Claude Opus 5, Claude Fable 5.1, and Grok 4.6 were added after the first results were seen. Their displays identify this later group explicitly. They use the same prompt, output allowance, and scoring contract, but reject the specified temperature field and run with provider-default sampling. Fable also rejects the thinking-disabled setting; Appendix F records the effective request for each family.

The primary parser accepts the specified JSON object without response repair. An invalid response receives an empty ranking, so formatting failures remain part of the evaluated pipeline. A separately labelled format-tolerant parse strips one surrounding code fence from the same recorded responses. It diagnoses a transport-format effect without replacing the primary scores or generating new responses.

Uncertainty intervals resample public source identifiers, retaining the items associated with each sampled source and the multiplicity of repeated draws. The displayed 95% intervals use 10,000 percentile-bootstrap replicates of the item-weighted statistic. They describe source-sampling variation for the retained runs; provider runs were not repeated to estimate sampling variability from generation.

5 RECOVERING THE WHOLE IMPACT SET

The paired scores reveal which systems assemble complete targets and which mainly recover parts of them. Figure 3 places complete-closure@2G beside Recall@2G for every retained system on the machine-validated candidate set. Both use the same reviewed prefix of each ranking, making the difference a property of the recovered set rather than the review allowance.

5.1 PARTIAL RECOVERY AND COMPLETION SEPARATE THE SYSTEMS

Llama 3.3 70B reaches complete-closure@2G 65% with Recall@2G 0.91, compared with BM25’s 20% and 0.66. The later-added Sonnet 5 reaches 82% and 0.95, the highest retained completion estimate. Thus the stronger language-model rankings improve both the amount of affected material found and the frequency of finishing the target. Their remaining gap between recall and completion identifies the endpoint that RippleKB measures: a nearly complete set still leaves a statement unreviewed.

Retrieval refinements illustrate why both scores are useful. BM25 with embedding reranking reaches 13% complete recovery and Recall@2G 0.68, compared with BM25’s 20% and 0.66. A higher partial-recall point estimate can accompany fewer completed targets.

The stricter review allowance tests whether affected units appear early. Recall@G is 0.37 for BM25, 0.78 for Llama, and 0.89 for the later-added Sonnet 5. These scores complement the paired larger-budget results above by showing how much target content occupies the first target-sized prefix. Appendix G retains all scores, intervals, and partition-level summaries.

5.2 THE MISSING UNIT CAN BE THE DIRECTLY EDITED SENTENCE

The depth summaries make the complete-recovery endpoint more concrete. BM25 and Llama recover the directly edited unit more often than downstream units, whereas the later-added Opus and Fable show the reverse ordering. Opus reaches Recall@2G 0.79 with complete-closure@2G 0%; its direct-unit recovery within the review prefix is zero. Its partial score reflects recovery elsewhere in the proposed dependency closure, while the absent direct unit prevents completion.

This distinction matters because the task includes the source sentence as well as consequences of its edit. The separate edit record tells the system what changes; it does not replace that sentence in the ranked answer. A ranking of downstream consequences alone can therefore be useful yet incomplete under the same target definition. Appendix J describes the supplied direct and downstream summaries and their scope.

Recovering many units can still leave the closure incomplete

All 150 candidates; primary parser; source-cluster intervals

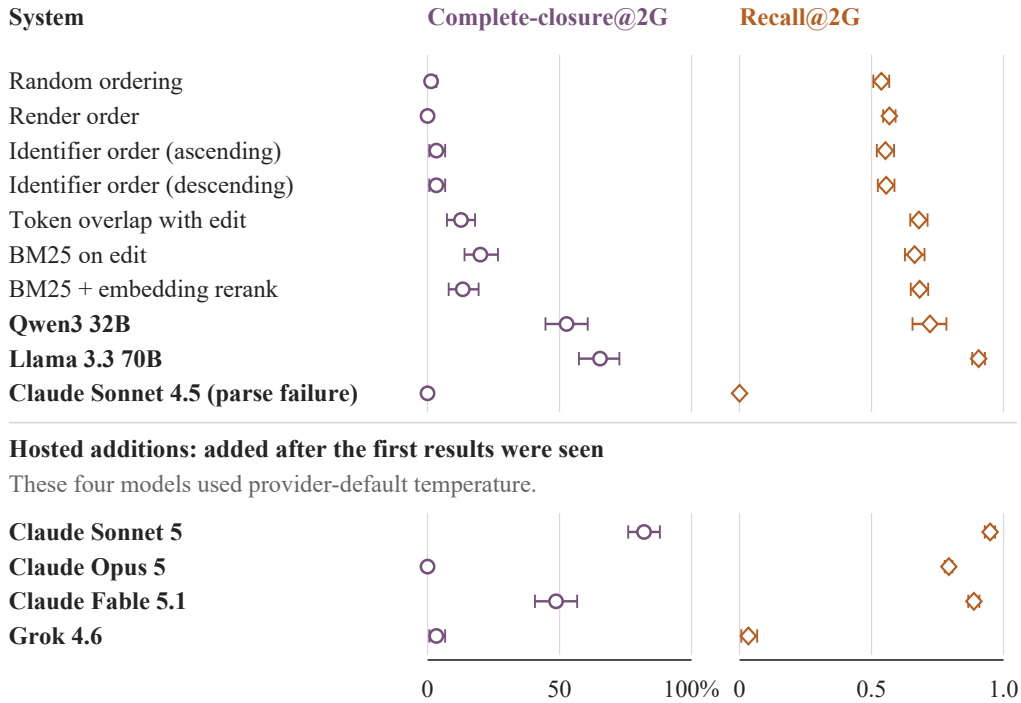


Figure 3: Complete recovery and recall under the primary response contract, with 95% source-cluster intervals. Sonnet 4.5’s zero scores reflect fenced responses; Opus’s zero completion reflects omission of the directly edited unit from the review prefix; Grok supplies only five parsed responses. The lower group contains models added after the first results were seen. Appendix H reports fence-removal rescoring of the same responses.

5.3 RESPONSE DELIVERY DETERMINES THE PRIMARY RANKING

The primary scores evaluate the delivered JSON response under the required parser. Sonnet 4.5 fences every response and therefore receives complete-closure@2G 0% with Recall@2G 0.00. The separately reported fence-stripping parse of those same responses gives 33% and 0.85. Qwen’s fenced responses similarly contribute failures under primary scoring; Appendix H places the two parses side by side.

This secondary measurement isolates a response-wrapper effect while holding the recorded model text fixed. It helps distinguish an unavailable ranking from a parsed ranking that omits affected units. The later-added Grok run supplies only five parsed responses under the 4,096-token output cap, reaching 3% complete recovery with Recall@2G 0.03. Empty and malformed responses remain in the full denominator.

Table 2: Removing a response wrapper changes access to the ranking. These are the two families with fenced responses; all scores retain 150 items. Only the surrounding fence is removed.

Model	Parse	Parsed	Complete@2G	Recall@2G
Qwen3 32B	Primary	120	53%	0.72
	Fence stripped	150	66%	0.90
Sonnet 4.5	Primary	0	0%	0.00
	Fence stripped	150	33%	0.85

Table 2 shows the same-response comparison directly. Qwen’s 30 fenced outputs become available to the scorer, while stripping Sonnet’s fences reveals a ranking with high partial recall and substantially fewer complete sets.

5.4 COPYING FIDELITY ANSWERS A SEPARATE QUESTION

Retrieval controls reproduce selected units exactly, yet their complete-recovery scores remain low. Among recovered affected units, Llama’s conditional verbatim exactness is 0.942 and Qwen’s is 0.995. The later-added Sonnet 5 reaches 0.998. These copying scores condition on finding an affected unit: a correctly selected identifier can carry altered text, while perfectly copied units can still leave other members of the impact set missing.

Numeric and named-entity fidelity include annotated spans in missing units in their denominators. Those measures therefore combine coverage with preservation, while conditional exactness measures copying after recovery. Appendix I reports them together with their respective denominators.

Table 3: Copying and coverage can diverge. Exact copying conditions on recovered affected units; the span columns retain all annotated affected spans, including missing units. Selected systems discussed in the text are shown; Appendix I gives every system.

System	Exact copying	Numeric spans	Entity spans
BM25	1.000	0.91	0.67
Qwen3 32B	0.995	0.75	0.72
Llama 3.3 70B	0.942	0.97	0.91
<i>Added after the first results were seen</i>			
Sonnet 5	0.998	0.94	0.95
Opus 5	0.999	0.28	0.79

Opus’s near-perfect conditional copying coexists with numeric-span preservation of 0.28 (Table 3). The conditional score describes its recovered units; the broader denominator also includes the annotated material it failed to return.

6 RELATED WORK

Finding affected artifacts. Change-impact analysis provides the closest task connection. Requirement analysis propagates relations among linked artifacts, while ProReFiCIA ranks requirements affected by a change rationale against expert labels (Goknil et al., 2016; Etezadi et al., 2025). RAID propagates edits through a compositional knowledge base and evaluates affected identifiers (Guo et al., 2026). RippleKB examines affected-unit recovery in linked free-form text, with unchanged source wording included in the output. The review allowance makes the ordering of affected and unaffected units part of the task. The target is the complete constructed set presented for review.

Updating text and model knowledge. FRUIT evaluates faithful Wikipedia updates, EditEval studies instruction-guided editing, and EditPropBench distinguishes required revisions from protected text (Logan IV et al., 2022; Dwivedi-Yu et al., 2024; Kruthof, 2026). FRESCO examines retrieval as corpora change (An et al., 2026). RippleKB evaluates the source statements selected for possible revision, before a system rewrites them. Knowledge-editing benchmarks instead examine consequences within a model: RippleEdits studies related changes, MQuAKE uses multi-hop questions, and logical-rule evaluations probe derived consequences (Cohen et al., 2024; Zhong et al., 2023; Moteu Ngoli et al., 2025). Locality evaluations and RippleBench examine unaffected cases and indirect effects of unlearning (Singh, 2026; Rinberg et al., 2025). Our affected and unaffected units remain jointly visible in the documents being ranked.

Constructing dependency-sensitive evaluations. Analyses of HotpotQA identify single-passage and disconnected-reasoning routes to correct answers (Chen & Durrett, 2019; Min et al., 2019; Trivedi et al., 2020). MuSiQue composes questions to require connected reasoning and supplies the public source records used here (Trivedi et al., 2022). Its constituent questions expose the intended reasoning

structure, while separate artifact-based models test how much of the benchmark can be solved through other routes. These studies motivate examining whether a dataset’s visible regularities reveal its labels. Such diagnostics matter when generation can make affected text look different from the unaffected alternatives. RippleKB tests identifier order, document positions, lexical overlap, original supporting facts, and composition regularity against its constructed targets. The released closure-review protocol separately asks whether the generated text supports the proposed dependencies and whether affected statements are missing.

Selecting evidence and preserving text. BERTSUM learns extractive selection; RECOMP, FILCO, and Provenance select or compress context for subsequent model use (Liu & Lapata, 2019; Xu et al., 2024; Wang et al., 2023; Chirkova et al., 2025). Symbolic rewriting produces compact fact lines, and matched-budget studies compare those representations with prose summaries in multi-hop question answering (Arbuzov et al., 2026; Bei et al., 2026). Cross-document evidence evaluation likewise studies responses supported by distributed information (Eletter et al., 2026). RippleKB’s exact-copy requirement retains each selected unit as an inspectable source object. Its scores distinguish selecting the right identifiers, completing the target, and preserving the returned text.

7 DISCUSSION

RippleKB makes a concrete retrieval failure visible: high partial recall can coexist with an unfinished impact set, including a ranking that misses the edited sentence itself.

Make completion an explicit objective. A ranking can be useful before it contains the complete impact set. The paired recovery measures show how much of a constructed target has been found and whether any member remains beyond the reviewed prefix. Reporting both connects an average retrieval score to the maintenance question that motivates the task. The direct and downstream summaries sharpen this distinction: missing the edited sentence can prevent completion even when many of its proposed consequences are recovered. Review budgets therefore belong alongside recovery scores when comparing systems.

Separate selection, delivery, and copying. The response contract creates an observable path from a model’s output to the source units a reviewer receives. The primary parser measures delivery under that contract. After a response is parsed, identifier recovery and exact copying answer separate questions. Conditional copying and selection-sensitive span preservation identify these different sources of loss.

Match construction checks to the claim. The proposed dependency graph makes the target inspectable, and the named controls test specific ways to recover it from visible regularities. Whether that key matches all statements changed by the edit remains a semantic question; the released human-review protocol specifies how to assess it.

RippleKB currently studies numeric edits in finite collections of constructed English documents derived from Wikipedia material. Within that setting, it exposes a review task that is easy to hide behind partial success: assembling every member of a proposed affected set and returning its source text intact. Its gold-size scoring allowance isolates ranking completeness; deploying the task in a review workflow would also require choosing a cutoff without access to the answer key.

REPRODUCIBILITY STATEMENT

The materials provide construction and metric specifications, aggregate result displays, response-configuration records, and hashes of referenced inputs. The figure scripts and fixed data extracts rebuild the manuscript displays. Appendix K identifies the item-level artifacts and executable components required for full replay. Timestamps are withheld from the anonymous version and restored at camera-ready; Appendix L retains the ordered changes.

ETHICS STATEMENT

The construction uses public source material under the documented upstream license and adds generated linked statements for evaluation. Redistribution requires source attribution, license notices, and a description of those modifications. The constructed documents are evaluation items rather than updates to the source encyclopedia.

AI USE STATEMENT

Generative models produce linked statements during benchmark construction, as described in Section 3. In this work, generative AI tools were used to polish the manuscript prose and assist with preparing explanatory illustrations. Generative AI tools were not used to design the model. All AI-assisted text and visual materials were reviewed and edited by the authors. The authors take responsibility for the final content of this paper, including all text, claims, and artifacts produced with AI assistance.

REFERENCES

- Sohyun An, Hayeon Lee, Shuibenyang Yuan, Chun-cheng Jason Chen, Cho-Jui Hsieh, Vijai Mohan, and Alexander Min. FRESCO: Benchmarking and Optimizing Re-rankers for Evolving Semantic Conflict in Retrieval-Augmented Generation. *arXiv preprint arXiv:2604.14227*, 2026. URL <https://arxiv.org/abs/2604.14227>.
- Mikhail L. Arbutov, Sisong Bei, Ziwei Dong, Dmitri Kalaev, and Alexey A. Shvets. Telegraph English: Semantic prompt compression via structured symbolic rewriting, 2026. URL <https://arxiv.org/abs/2605.04426v1>.
- Sisong Bei, Mikhail L. Arbutov, Ziwei Dong, Dmitri Kalaev, and Alexey Shvets. Context compression is not one thing: Readable symbolic re-expression vs. coherent summary at matched budget, 2026. URL <https://arxiv.org/abs/2606.14875v1>.
- A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-Based Improvements for Inference with Clustered Errors. *Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414. URL <https://doi.org/10.1162/rest.90.3.414>.
- Jifan Chen and Greg Durrett. Understanding Dataset Design Choices for Multi-hop Reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4026–4032, 2019. doi: 10.18653/v1/N19-1405. URL <https://aclanthology.org/N19-1405/>.
- Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: efficient and robust context pruning for retrieval-augmented generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5e956fef0946dc1e39760f94b78045fe-Abstract-Conference.html.
- Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024. doi: 10.1162/tacl_a_00644. URL <https://aclanthology.org/2024.tacl-1.16/>.

- Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. EditEval: An Instruction-Based Benchmark for Text Improvements. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pp. 69–83, 2024. doi: 10.18653/v1/2024.conll-1.7. URL <https://aclanthology.org/2024.conll-1.7/>.
- Saadeldine Eletter, Ruihong Zeng, Yuxia Wang, Maxim Panov, Aleksandr Rubashevskii, and Preslav Nakov. MIRAGE: Defending Long-Form RAG Against Misinformation Pollution. *arXiv preprint arXiv:2607.05069*, 2026. URL <https://arxiv.org/abs/2607.05069>.
- Romina Etezadi, Sallam Abualhaija, Chetan Arora, and Lionel Briand. LLM-Driven Cost-Effective Requirements Change Impact Analysis. *arXiv preprint arXiv:2511.00262*, 2025. URL <https://arxiv.org/abs/2511.00262>.
- Arda Goknil, Ivan Kurtev, and Klaas van den Berg. A Rule-Based Change Impact Analysis Approach in Software Architecture for Requirements Changes. *arXiv preprint arXiv:1608.02757*, 2016. URL <https://arxiv.org/abs/1608.02757>.
- Jiajing Guo, Xueming Li, Jorge Piazentin Ono, Wenbin He, and Liu Ren. Scaling Expert Feedback with Reflective Edit Propagation in Compositional Knowledge Bases. In *Proceedings of the ACM Conference on AI and Agentic Systems*, 2026. doi: 10.1145/3786335.3813201. URL <https://arxiv.org/abs/2606.05023>.
- Garvin Kruthof. EditPropBench: Measuring Factual Edit Propagation in Scientific Manuscripts. *arXiv preprint arXiv:2605.02083*, 2026. URL <https://arxiv.org/abs/2605.02083>.
- Yang Liu and Mirella Lapata. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3730–3740, 2019. doi: 10.18653/v1/D19-1387. URL <https://aclanthology.org/D19-1387/>.
- Robert L. Logan IV, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. FRUIT: Faithfully Reflecting Updated Information in Text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3670–3686, 2022. doi: 10.18653/v1/2022.naacl-main.269. URL <https://aclanthology.org/2022.naacl-main.269/>.
- Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. Compositional Questions Do Not Necessitate Multi-hop Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4249–4257, 2019. doi: 10.18653/v1/P19-1416. URL <https://aclanthology.org/P19-1416/>.
- Tatiana Moteu Ngoli, N’Dah Jean Kouagou, Hamada M. Zahera, and Axel-Cyrille Ngonga Ngomo. Benchmarking Knowledge Editing using Logical Rules. In *The Semantic Web – ISWC 2025*, volume 16141 of *Lecture Notes in Computer Science*, pp. 41–56. Springer, 2025. doi: 10.1007/978-3-032-09530-5_3. URL https://link.springer.com/chapter/10.1007/978-3-032-09530-5_3.
- Roy Rinberg, Usha Bhalla, Igor Shilov, Flavio P. Calmon, and Rohit Gandikota. RippleBench: Capturing Ripple Effects Using Existing Knowledge Repositories. *arXiv preprint arXiv:2512.04144*, 2025. URL <https://arxiv.org/abs/2512.04144>.
- Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 2009. doi: 10.1561/15000000019. URL <https://doi.org/10.1561/15000000019>.
- Aditya Pratap Singh. On Scope Classification and Current Knowledge-Editing Benchmarks: A Negative Result, with INLAY as a Gradient-Free Case Study. *arXiv preprint arXiv:2608.26292*, 2026. URL <https://arxiv.org/abs/2608.26292>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Is Multihop QA in DiRe Condition? Measuring and Reducing Disconnected Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8846–8863,

2020. doi: 10.18653/v1/2020.emnlp-main.712. URL <https://aclanthology.org/2020.emnlp-main.712/>.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL <https://aclanthology.org/2022.tacl-1.31/>.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. Learning to Filter Context for Retrieval-Augmented Generation. *arXiv preprint arXiv:2311.08377*, 2023. URL <https://arxiv.org/abs/2311.08377>.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=mlJLVigNHp>.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL <https://aclanthology.org/D18-1259/>.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL <https://aclanthology.org/2023.emnlp-main.971/>.

A STATISTICAL PROCEDURE

The result tables describe all 150 machine-validated candidates, including the construction-development, selection, and report partitions. The initial baseline configuration preceded candidate scoring; no settings were tuned on these partitions. Later hosted additions followed inspection of the first results and are identified separately. The earlier construction-control diagnostics remain distinct from this baseline comparison.

The supplied uncertainty procedure resamples public source identifiers with replacement, retaining all items associated with each sampled source and the multiplicity of repeated draws. Each replicate computes the item-weighted macro statistic; 10,000 replicates supply the displayed 95% percentile intervals (Cameron et al., 2008). The numeric seed is retained in the release configuration under its designated label. These intervals quantify source-sampling variation under the retained runs; the tables contain no repeat-provider-run uncertainty. The accompanying manifests bind the result exports to the scoring configuration. The item-level records and executable scorer are absent from this handoff, so this manuscript reports the supplied intervals without independently rerunning them.

The construction checks and candidate selection used hidden construction labels before model evaluation. The final all-candidate comparison is descriptive; the partition names record construction assignments.

B CONSTRUCTION SCHEMA AND VALIDATION

B.1 EDIT, SOURCE UNIT, AND HIDDEN GRAPH

The edit record specifies the source unit, an exact old substring, a replacement substring, and a visible description. The edit record itself is separate from the candidate units. The edited source sentence belongs to the constructed dependency closure. Its hidden edge originates at the edit record, so its recorded depth is one. Other affected sentences receive depths from their shortest paths from that record. This convention distinguishes the directly edited source sentence from downstream consequences.

The public representation contains document identifiers, titles, unit identifiers, canonical text, and text hashes. Hidden construction metadata stores source provenance, graph edges, template membership, original supporting-fact labels, closure membership, and numeric and entity spans. Systems receive the public representation and edit description. The private fields determine construction checks and later scoring.

B.2 SELECTION AND MECHANICAL CHECKS

Source assignment precedes generation and assigns an ordered reserve list to each structural coordinate. The selector takes the first mechanically valid completion in that predetermined list. The aggregate records 1,800 attempted sources, 1,564 mechanical passes, and 150 selected candidates, with 12 reserves per coordinate. Each source eligible for inference receives one completion; failed preflight checks reject a source before generation. The selection rule uses neither baseline scores nor human verdicts. The selected candidates follow the structural schedule rather than a random sample of all mechanical passes.

Validation checks the planned slots, graph edges, documents, candidate counts, source grounding, and exact edit substring. It also checks reachability, unaffected units, cross-document consequences, original-support coverage, and canonical text and span hashes. Affected downstream text is composed from a named entity followed by its predicate, with the entity anchored at the start. Unaffected text is generated without that prefix requirement. Internal roles are checked before opaque public identifiers are assigned. Document order and unit order are shuffled independently. The released closure-review protocol specifies how to assess textual support for edges, unintended dependencies, and affected-set completeness.

B.3 DEPENDENCY SHAPES

Table 4 gives the selected candidate counts for the constructed graph families. Each family contains a branch, and some contain a merge. A merge requires the stated target to follow from each specified parent under the rendered facts. Table 5 summarizes the corresponding document counts, impact-set sizes, and depths.

Table 4: Dependency-shape counts on the machine-validated candidate set.

Shape	Candidates
Branched merge	17
Chain with branch	23
Chain with late branch	19
Deep branch	19
Diamond with tail	18
Unequal-depth branches	19
Fork with tails	19
Three branches	16

Table 5: Candidate composition by document count, impact size, and maximum depth on the machine-validated candidate set.

Property	Value	Candidates
Documents	3	75
Documents	4	75
Affected units	4	23
Affected units	5	56
Affected units	6	38
Affected units	7	33
Maximum depth	3	42
Maximum depth	4	72
Maximum depth	5	36

The schedule represents every listed family. Its most frequent shape accounts for the displayed 15.3% of candidates, and its most frequent affected-position pattern accounts for 1.3%. The composition check tests representation, concentration, coverage, and identifier uniqueness. It does not produce a ranked list or a recovery score.

C SHORTCUT CONTROLS

C.1 CONTROL DEFINITIONS

The identifier control sorts public identifiers in ascending and descending order. For each metric, its diagnostic reports the larger score across these orderings. The document-position control learns affected-unit frequencies from public structural features on development candidates. Its fixed predictor is evaluated separately on selection and report candidates. There is no pooled document-position score in the supplied report.

The lexical controls rank by edit-token overlap or BM25 similarity to the edit (Robertson & Zaragoza, 2009). The reported lexical maxima are computed separately for complete recovery and recall. The same row can therefore combine complete recovery from BM25 with recall from token overlap. These component maxima summarize a control family rather than an individual ranking method.

The supporting-fact control receives the hidden original-support labels as privileged information. It ranks those units first and orders the remaining units randomly using the fixed seed. It is a

diagnostic of reliance on the source question’s support, rather than a deployable retrieval system. The composition check examines graph-family and position frequencies separately from these ranking controls.

C.2 CANDIDATE RESULTS AND THRESHOLDS

Table 6 pairs complete recovery with recall for every displayed control evaluation. The pooled rows refer to all 150 machine-validated candidates. The random row is the earlier seeded construction diagnostic, whose ordering differs from the retained random baseline in Appendix G.

Table 6: Construction-control results on the machine-validated candidate set. Component maxima can come from different rankings.

Control	Split	Complete (%)	Recall@2G
Identifier max.	all 150	3.3%	0.56
Identifier max.	development (30)	10.0%	0.57
Identifier max.	selection (30)	0.0%	0.56
Identifier max.	report (90)	5.6%	0.57
Document position	selection (30)	3.3%	0.60
Document position	report (90)	3.3%	0.63
Lexical maxima	all 150	20.0%	0.68
Lexical maxima	development (30)	13.3%	0.68
Lexical maxima	selection (30)	26.7%	0.64
Lexical maxima	report (90)	20.0%	0.69
Supporting facts	all 150	4.7%	0.63
Supporting facts	development (30)	6.7%	0.58
Supporting facts	selection (30)	6.7%	0.63
Supporting facts	report (90)	3.3%	0.64
Random ordering	all 150	4.0%	0.57

Lexical matching recovers some complete constructed closures, and its selection-split maximum exceeds its pooled value. The registered thresholds are decision rules for detecting the specified construction shortcuts. Passing them establishes the stated checks; semantic acceptance requires the separate review procedure. Table 7 records the ranking thresholds applied to every required evaluation.

Table 7: Registered ranking thresholds for complete recovery and recall. Each required split evaluation must remain below both thresholds.

Control	Complete (%)	Recall@2G
Identifier order	< 35.0	< 0.80
Document position	< 35.0	< 0.80
Lexical matching	< 60.0	< 0.90
Supporting facts	< 35.0	< 0.75

The supporting-fact checks additionally require affected units outside the source question’s support and across document boundaries. The generated report gives the non-supporting share of affected units as 81.9%, against the registered minimum of 60.0%. Every item must contain at least two affected units outside the edited document.

C.3 WITHDRAWN CONSTRUCTION

An earlier construction gave all 44 development items the same 14-unit structure and four ascending affected identifiers. Identifier order recovered every affected unit without using its text. Those items were withdrawn before audit distribution, and the replacement construction assigned opaque

identifiers after content validation. Later development revisions addressed lexical leakage, coverage failures, and structured emission, while retaining the registered shortcut thresholds. The surviving candidate diagnostics evaluate that revised construction.

D RELEASED CLOSURE-REVIEW PROTOCOL

The protocol specifies a review of the constructed answer key against the visible documents. It separates independent discovery of affected units from inspection of the proposed dependencies.

During blind recovery, two independent reviewers receive identical public documents and the visible edit. Each selects affected units, gives rationales, checks omissions, and records ambiguity; the proposed closure remains hidden. Closure review then reveals the proposed membership and graph edges for independent unit-by-unit and edge-by-edge assessment. Reviewers check textual support, missing affected statements, unintended dependencies, and ambiguity. Disagreements and unresolved defects enter joint adjudication after both reviews are committed.

Table 8: Released closure-review procedure. The stages assess target membership and dependency support against the documents.

Step	Evidence required
Blind recovery	Independent selections, rationales, and omission checks.
Closure review	Textual support for membership and dependency edges.
Adjudication	Resolution of membership, edge, and ambiguity questions.

The acceptance rule requires a valid final closure, canonical annotations, source isolation, and no unresolved membership or dependency question. The procedure preserves independent returns beside the adjudicated outcome and requires attributable effort records.

E METRIC DEFINITIONS

E.1 RANKED SELECTION

Let U_i be the candidate units and $G_i \subseteq U_i$ the target affected units for item i . The review cutoff is $K_i = \min(|U_i|, 2|G_i|)$. Let $R_i(k)$ contain distinct valid affected identifiers whose first submitted occurrence lies within the first k positions. Recall at the review cutoff is $|R_i(K_i)|/|G_i|$. Complete-closure@2G averages the indicator $\mathbf{1}\{R_i(K_i) = G_i\}$ over items. Recall@G uses the smaller cutoff $\min(|U_i|, |G_i|)$.

Unknown identifiers, duplicates, and missing required fields consume positions without receiving selection credit. A valid affected identifier can receive selection credit when its text is inexact. Malformed or missing item output is scored as an empty list. Short lists receive no credit for their missing positions. Identifiers are never inferred from approximate text matches.

Precision at the review cutoff is $P_i = |R_i(K_i)|/K_i$. F-measure combines that precision and recall by their harmonic mean, with zero assigned when both are zero. Average precision sums the precision at each first recovered affected identifier and divides by $|G_i|$. Missing affected units contribute zero. Each item-level statistic is macro-averaged over the evaluation items.

Review burden records the first rank attaining the registered target recall, normalized by impact-set size. If that target is unreached within $|U_i|$ positions, the recorded rank is $|U_i| + 1$ before normalization. The associated success proportion distinguishes unrecovered items from successful review. An all-candidate ranking receives the same prefix evaluation as any other ranking. Its unrestricted recall can reach complete recovery, while its cutoff scores and burden still depend on ordering.

E.2 TEXT AND SPAN FIDELITY

Canonical text comparison decodes the submitted JSON string, converts transport-level CRLF line endings to LF, and compares the resulting UTF-8 bytes. Whitespace, punctuation, Unicode form, and capitalization otherwise remain unchanged. Verbatim Impact Recall@2G divides correctly copied affected units within the cutoff by all affected units. Conditional exactness divides correctly copied recovered units by all recovered affected units, micro-aggregated over the evaluation split. If no affected identifier is recovered, the conditional score is zero.

Numeric and named-entity annotations identify exact byte intervals in canonical text. A span is preserved only when its affected identifier is recovered within the cutoff and identical bytes occupy the annotated interval. Selection-sensitive fidelity divides preserved spans by all annotated affected spans. Conditional preservation instead divides by annotated spans in recovered affected units. The result export includes selection-sensitive span fidelity and conditional-preservation fields. The displayed fidelity table reports the former; conditional verbatim exactness uses recovered affected units as its denominator. Per-item annotation counts required for full replay are absent from the supplied aggregate displays. Moved spans, numeric reformatting, aliases, and paraphrases fail this exact preservation test.

F BASELINE SPECIFICATIONS AND RETAINED RUNS

The retained initial families comprise six deterministic rankings, BM25 with embedding reranking, Qwen3 32B, Llama 3.3 70B, and Claude Sonnet 4.5. The reranker uses Titan Text Embeddings V2; its supplied configuration specifies BM25 retrieval followed by embedding similarity. Every model receives the same public edit and candidate documents. The system prompt requires all candidate identifiers and exact source text:

```
You rank source units affected by a visible edit.
Use only the edit and documents provided. Hidden dependency edges and
↔ gold
membership are unavailable. Rank every candidate unit from most to least
likely affected. Copy each unit's text byte-for-byte from the input.
Return one JSON object and no markdown or commentary:
{"ranking":[{"unit_id":"opaque-id","text":"exact source text"}]}
```

The requested decoding settings are temperature zero, top-p one, and a 4,096-token output cap. Exact endpoint identifiers, request records, prompt hashes, and scorer hashes remain in the accompanying configurations; calendar-bearing identifier suffixes are omitted from this PDF. The initial configuration specified selection and report evaluation; the revised comparison uses all 150 machine-validated candidates without tuning on any partition.

Each retained scoring run covers its assigned candidate items once. The campaign also includes provider-compatibility reruns: the initial Llama result was replaced, and the Sonnet 4.5 result was replaced twice. The final result manifest identifies which run contributes each row and preserves the earlier manifests. The pipeline's no-repair primary scoring rule applies to the retained responses.

F.1 HOSTED FAMILIES ADDED AFTER THE FIRST RESULTS WERE SEEN

Claude Sonnet 5, Claude Opus 5, Claude Fable 5.1, and Grok 4.6 were added under the same prompt, output cap, parser, and bootstrap procedure. Their provider request schemas reject the specified temperature field; neither temperature nor top-p is sent, so provider-default sampling applies. Table 9 gives the effective settings, including the thinking-field exception.

Table 9: Provider settings for families added after the first results were seen. All use the same output cap; sampling follows provider defaults.

Family	Sampling	Thinking field
Claude Sonnet 5	Provider default	Disabled
Claude Opus 5	Provider default	Disabled
Claude Fable 5.1	Provider default	Omitted
Grok 4.6	Provider default	Omitted

Fable’s request schema also rejects a disabled-thinking field.

G COMPLETE AND PARTIAL RECOVERY

The following scorecards copy the supplied displays at their published precision. Complete-closure@2G and Recall@2G always appear together; Recall@G uses the stricter cutoff. Brackets give the supplied source-cluster intervals. Hosted additions are grouped by their later introduction and provider-default sampling.

Table 10: Primary recovery with 95% intervals on the machine-validated candidate set.

System	Complete@2G	Recall@2G	Recall@G
Random ordering	1% [0, 3]	0.54 [0.51, 0.57]	0.27 [0.24, 0.30]
Render order	0% [0, 0]	0.57 [0.54, 0.59]	0.28 [0.26, 0.30]
Identifier ascending	3% [1, 7]	0.55 [0.52, 0.59]	0.27 [0.24, 0.29]
Identifier descending	3% [1, 7]	0.56 [0.52, 0.59]	0.29 [0.26, 0.32]
Token overlap	13% [7, 18]	0.68 [0.65, 0.71]	0.39 [0.36, 0.43]
BM25	20% [14, 27]	0.66 [0.63, 0.70]	0.37 [0.34, 0.40]
BM25 + embeddings	13% [8, 19]	0.68 [0.65, 0.71]	0.43 [0.39, 0.46]
Qwen3 32B	53% [45, 61]	0.72 [0.66, 0.78]	0.64 [0.58, 0.70]
Llama 3.3 70B	65% [57, 73]	0.91 [0.88, 0.93]	0.78 [0.74, 0.81]
Claude Sonnet 4.5	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
<i>Added after the first results were seen; provider-default sampling</i>			
Claude Sonnet 5	82% [76, 88]	0.95 [0.93, 0.97]	0.89 [0.86, 0.92]
Claude Opus 5	0% [0, 0]	0.79 [0.78, 0.81]	0.75 [0.72, 0.78]
Claude Fable 5.1	49% [41, 57]	0.89 [0.87, 0.91]	0.85 [0.82, 0.87]
Grok 4.6	3% [1, 7]	0.03 [0.01, 0.07]	0.03 [0.01, 0.06]

G.1 DEVELOPMENT (30 CANDIDATES)

Table 11: Development (30 candidates) on the machine-validated candidate set.

System	Complete@2G	Recall@2G	Recall@G
Random ordering	0% [0, 0]	0.51 [0.43, 0.58]	0.29 [0.22, 0.35]
Render order	0% [0, 0]	0.55 [0.50, 0.60]	0.29 [0.24, 0.33]
Identifier ascending	10% [0, 20]	0.57 [0.49, 0.65]	0.27 [0.21, 0.32]
Identifier descending	0% [0, 0]	0.54 [0.46, 0.62]	0.29 [0.22, 0.36]
Token overlap	7% [0, 17]	0.68 [0.62, 0.74]	0.41 [0.34, 0.47]
BM25	13% [3, 27]	0.67 [0.59, 0.74]	0.37 [0.31, 0.44]
BM25 + embeddings	10% [0, 23]	0.66 [0.59, 0.73]	0.39 [0.32, 0.47]
Qwen3 32B	57% [40, 73]	0.71 [0.57, 0.85]	0.55 [0.41, 0.69]
Llama 3.3 70B	70% [53, 87]	0.91 [0.85, 0.96]	0.74 [0.65, 0.83]
Claude Sonnet 4.5	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]

Table 11 (continued)

System	Complete@2G	Recall@2G	Recall@G
<i>Added after the first results were seen; provider-default sampling</i>			
Claude Sonnet 5	80% [67, 93]	0.95 [0.90, 0.98]	0.87 [0.80, 0.94]
Claude Opus 5	0% [0, 0]	0.80 [0.77, 0.82]	0.72 [0.63, 0.80]
Claude Fable 5.1	50% [33, 67]	0.89 [0.84, 0.94]	0.83 [0.77, 0.90]
Grok 4.6	7% [0, 17]	0.07 [0.00, 0.17]	0.07 [0.00, 0.17]

G.2 SELECTION (30 CANDIDATES)

Table 12: Selection (30 candidates) on the machine-validated candidate set.

System	Complete@2G	Recall@2G	Recall@G
Random ordering	7% [0, 17]	0.59 [0.52, 0.67]	0.30 [0.23, 0.36]
Render order	0% [0, 0]	0.56 [0.51, 0.61]	0.27 [0.23, 0.32]
Identifier ascending	0% [0, 0]	0.56 [0.51, 0.62]	0.30 [0.23, 0.37]
Identifier descending	0% [0, 0]	0.53 [0.47, 0.58]	0.28 [0.23, 0.33]
Token overlap	10% [0, 23]	0.64 [0.57, 0.71]	0.36 [0.31, 0.42]
BM25	27% [13, 43]	0.64 [0.55, 0.73]	0.33 [0.27, 0.40]
BM25 + embeddings	27% [13, 43]	0.77 [0.71, 0.83]	0.51 [0.43, 0.58]
Qwen3 32B	50% [33, 67]	0.65 [0.49, 0.80]	0.62 [0.47, 0.77]
Llama 3.3 70B	60% [43, 77]	0.91 [0.87, 0.95]	0.79 [0.72, 0.86]
Claude Sonnet 4.5	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
<i>Added after the first results were seen; provider-default sampling</i>			
Claude Sonnet 5	70% [53, 87]	0.93 [0.89, 0.97]	0.89 [0.82, 0.95]
Claude Opus 5	0% [0, 0]	0.79 [0.75, 0.81]	0.74 [0.68, 0.79]
Claude Fable 5.1	40% [23, 57]	0.86 [0.81, 0.91]	0.83 [0.77, 0.89]
Grok 4.6	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]

G.3 REPORT (90 CANDIDATES)

Table 13: Report (90 candidates) on the machine-validated candidate set.

System	Complete@2G	Recall@2G	Recall@G
Random ordering	0% [0, 0]	0.53 [0.49, 0.56]	0.25 [0.21, 0.29]
Render order	0% [0, 0]	0.57 [0.54, 0.61]	0.28 [0.26, 0.30]
Identifier ascending	2% [0, 6]	0.54 [0.50, 0.59]	0.26 [0.22, 0.29]
Identifier descending	6% [1, 11]	0.57 [0.53, 0.61]	0.29 [0.26, 0.33]
Token overlap	16% [9, 23]	0.69 [0.65, 0.73]	0.40 [0.35, 0.44]
BM25	20% [12, 29]	0.67 [0.62, 0.72]	0.38 [0.34, 0.42]
BM25 + embeddings	10% [4, 17]	0.66 [0.61, 0.70]	0.41 [0.37, 0.46]
Qwen3 32B	52% [42, 62]	0.75 [0.67, 0.82]	0.68 [0.60, 0.75]
Llama 3.3 70B	66% [56, 76]	0.90 [0.87, 0.93]	0.78 [0.73, 0.83]
Claude Sonnet 4.5	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
<i>Added after the first results were seen; provider-default sampling</i>			
Claude Sonnet 5	87% [79, 93]	0.96 [0.93, 0.98]	0.89 [0.85, 0.93]
Claude Opus 5	0% [0, 0]	0.79 [0.77, 0.81]	0.76 [0.73, 0.79]
Claude Fable 5.1	51% [41, 61]	0.90 [0.87, 0.92]	0.85 [0.82, 0.89]
Grok 4.6	3% [0, 8]	0.03 [0.00, 0.08]	0.03 [0.00, 0.07]

H FORMAT-TOLERANT PARSE (SECONDARY)

The secondary parser strips one leading code fence and one trailing code fence from the same saved response. It preserves the enclosed content and makes no new model call. The primary parser remains the no-repair evaluation. The saved transformation changes Qwen’s 30 fenced responses and all 150 Sonnet 4.5 responses; the other displayed families have no fence removal. Table 14 retains each primary row beside its secondary counterpart.

Table 14: Format-tolerant parse (secondary), alongside primary results, on the machine-validated candidate set.

Parse	Parsed	Complete@2G	Recall@2G	Recall@G
Qwen3 32B				
Primary	120	53% [45, 61]	0.72 [0.66, 0.78]	0.64 [0.58, 0.70]
Fence stripped	150	66% [58, 73]	0.90 [0.87, 0.93]	0.80 [0.76, 0.83]
Llama 3.3 70B				
Primary	150	65% [57, 73]	0.91 [0.88, 0.93]	0.78 [0.74, 0.81]
Fence stripped	150	65% [57, 73]	0.91 [0.88, 0.93]	0.78 [0.74, 0.81]
Claude Sonnet 4.5				
Primary	0	0% [0, 0]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
Fence stripped	150	33% [26, 41]	0.85 [0.83, 0.87]	0.79 [0.76, 0.82]
<i>Added after the first results were seen; provider-default sampling</i>				
Claude Sonnet 5				
Primary	150	82% [76, 88]	0.95 [0.93, 0.97]	0.89 [0.86, 0.92]
Fence stripped	150	82% [76, 88]	0.95 [0.93, 0.97]	0.89 [0.86, 0.92]
Claude Opus 5				
Primary	150	0% [0, 0]	0.79 [0.78, 0.81]	0.75 [0.72, 0.78]
Fence stripped	150	0% [0, 0]	0.79 [0.78, 0.81]	0.75 [0.72, 0.78]
Claude Fable 5.1				
Primary	150	49% [41, 57]	0.89 [0.87, 0.91]	0.85 [0.82, 0.87]
Fence stripped	150	49% [41, 57]	0.89 [0.87, 0.91]	0.85 [0.82, 0.87]
Grok 4.6				
Primary	5	3% [1, 7]	0.03 [0.01, 0.07]	0.03 [0.01, 0.06]
Fence stripped	5	3% [1, 7]	0.03 [0.01, 0.07]	0.03 [0.01, 0.06]

I COPYING AND ANNOTATED-SPAN PRESERVATION

Conditional exactness is micro-aggregated over recovered affected units; a system recovering none receives zero. Numeric and named-entity fidelity divide preserved spans by all annotated affected spans, so omissions reduce their scores. These columns therefore measure different denominators rather than alternative estimates of one accuracy.

Table 15: Copying and span preservation on the machine-validated candidate set.

System	Exact copying	Numeric spans	Entity spans
Random ordering	1.000	0.56	0.55
Render order	1.000	0.58	0.57
Identifier ascending	1.000	0.54	0.55
Identifier descending	1.000	0.59	0.56
Token overlap	1.000	0.85	0.68
BM25	1.000	0.91	0.67
BM25 + embeddings	1.000	0.77	0.68

Table 15 (continued)

System	Exact copying	Numeric spans	Entity spans
Qwen3 32B	0.995	0.75	0.72
Llama 3.3 70B	0.942	0.97	0.91
Claude Sonnet 4.5	0.000	0.00	0.00
<i>Added after the first results were seen; provider-default sampling</i>			
Claude Sonnet 5	0.998	0.94	0.95
Claude Opus 5	0.999	0.28	0.79
Claude Fable 5.1	0.999	0.68	0.89
Grok 4.6	1.000	0.02	0.03

J DIRECT AND DOWNSTREAM RECOVERY

The supplied depth export separates the edited source unit, one edge from the visible edit record, from the deeper affected units. Each method has 150 eligible items for both summaries. The scores concern recovery within the same review prefix as the full-set measurements.

The export records higher direct than downstream recovery for BM25 and Llama, and the reverse for the later-added Opus and Fable. Opus’s zero direct-recovery score concerns only positions within the reviewed prefix. Empty or malformed responses contribute zero in both summaries. The machine-readable release retains every exported mean and eligible-item count alongside the complete-closure scorecards.

The broader registered difficulty slices include document boundaries, original supporting facts, impact size, candidate count, lexical overlap, and annotated spans. No item-level error census or manual taxonomy counts are supplied for those categories. The present interpretation uses the exported depth summaries, parser outcomes, and copying measurements.

K SOURCE ISOLATION, LICENSING, AND REPRODUCIBILITY

The source is the public MuSiQue release, whose questions combine evidence across Wikipedia-derived passages (Trivedi et al., 2022). The supplied record identifies a pinned source revision and archive/member hashes. Source identifiers used by another in-review project were excluded before split assignment, as recorded in the supplementary source-isolation disclosure. The separation record reports disjoint construction-development, selection, and report identifiers, with no transfer of derived translations, model outputs, annotations, or empirical claims.

Source-identifier separation defines item provenance; shared passages and prior model exposure remain unassessed. MuSiQue’s Creative Commons Attribution license requires attribution and identification of modifications in source-derived redistribution. New benchmark fields remain distinguishable from upstream material.

Available materials contain aggregate results, generated displays, design specifications, audit protocols, request records, and hashes of the referenced inputs. Item-level baseline responses, full private source mappings, executable construction and scoring code, and the original source-isolation records are needed for independent replay. The supplied aggregates permit inspection and display reproduction, while exact result recomputation requires those additional artifacts.

L CONSTRUCTION DECISIONS AFFECTING INTERPRETATION

Recorded pilots failed lexical and role/template checks, and one graph family lacked passing candidates. Revisions introduced edit-token restrictions, coverage changes, and structured entity emission. The construction-control report records access to selection and report labels before baseline outputs were available. Baseline evaluation used all candidates without tuning on their partitions. Fence-removal rescoring and the additional hosted families followed inspection of the first baseline results. Appendix C documents the withdrawn construction and final controls; Appendix F records retained runs and effective settings. The complete construction record remains in the reproducibility archive.