ENRICHING DOCUMENT KNOWLEDGE BASES THROUGH CONTEXT PROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding real-world document collections, such as conversations or codebases, requires more than retrieving isolated facts: it involves reconciling information scattered across documents and forming a coherent view of entities and events. This can be challenging because authors often assume shared context, using *implicit cross-document references* whose interpretation depends on information in other documents. To isolate this challenge, we introduce XRefQA, a diagnostic benchmark of small synthetic document collections in which answering questions requires resolving these references and combining evidence from multiple documents. Even Claude Opus 4.6 achieves only 7% accuracy on XRefQA using standard retrieval-augmented generation (RAG). To address this limitation, we propose *context propagation*, which iteratively enriches each document with source-grounded annotations using evidence retrieved from related documents. Enriched documents inform subsequent rounds, allowing resolved references and related facts to propagate across the collection. After two propagation rounds, the weaker Claude Sonnet 4.6 achieves 80% accuracy with RAG over enriched documents. Beyond implicit-reference resolution, the method also improves multi-hop question answering on MuSiQue, 2WikiMultiHopQA, and HotpotQA, as well as fact discovery and source attribution on SummHay. These results suggest that making cross-document connections explicit during preprocessing helps downstream systems retrieve, combine, and attribute distributed evidence.

1 INTRODUCTION

We aim to build AI agents that understand the details of the projects they work on, rather than merely retrieve isolated facts from project documents. Such understanding requires integrating information across sources, resolving ambiguity, and forming a coherent view of entities, events, and decisions.

A common approach to enabling such understanding is to place project context directly into the model prompt, an approach that has become increasingly attractive as modern large language models support ever larger context windows. However, performance can degrade as more information is included, while latency and cost constrain how much context can be processed effectively (Liu et al., 2024; Levy et al., 2024; Li et al., 2025). Retrieval-augmented generation (RAG) (Lewis et al., 2020; Ram et al., 2023) offers a more scalable alternative by retrieving a small number of relevant documents or snippets at inference time instead of including the entire corpus. While RAG is effective in many settings, it faces a fundamental limitation: retrieved snippets often lack sufficient context to be interpreted correctly (Choi et al., 2021; Chen et al., 2024; Anthropic, 2024).

One source of this difficulty is the use of *implicit cross-document references*: mentions of entities or events that require information from other documents to be resolved (Bagga & Baldwin, 1998; Cybulska & Vossen, 2014). Such references arise naturally in project communication, where authors use shorthand expressions such as "the boss" or "yesterday's meeting," assuming that their collaborators already know whom or what they mean. What is clear to these collaborators may therefore be ambiguous to a system reading a passage in isolation (Dakle & Moldovan, 2020). For example, a message stating that "the boss's birthday is on 12 May" can answer a question about Maria's birthday, but the connection becomes clear only when another document identifies Maria as the boss. Resolving these references is therefore important both for recognizing which documents are relevant and for interpreting the evidence they contain (Chen et al., 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

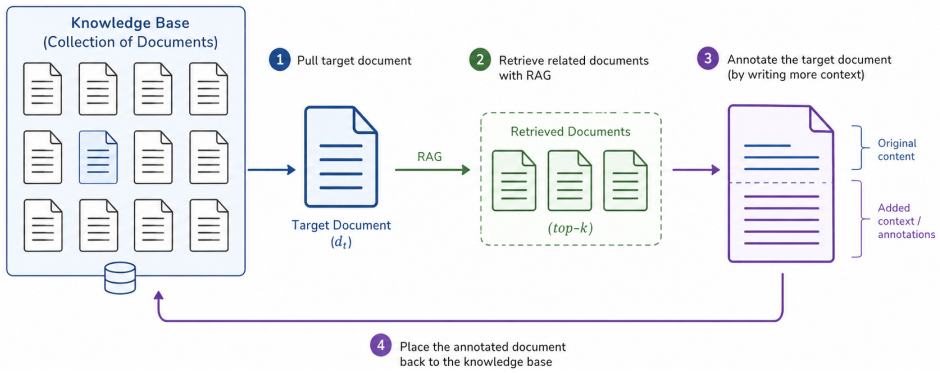


Figure 1: Overview of context propagation. For each document, we retrieve evidence from the same collection and add inline annotations that resolve implicit references and supply context. Enriched documents then inform subsequent rounds and support downstream retrieval and question answering.

Yet widely used multi-document question-answering benchmarks (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022) generally focus on combining evidence across passages whose connecting entities are named explicitly, rather than resolving such implicit references. Cross-document coreference research has instead addressed one part of the problem: linking mentions of the same entities or events across documents (Bagga & Baldwin, 1998; Cattan et al., 2021). Benchmarks such as ECB+ (Cybulska & Vossen, 2014) and CEREC (Dakle & Moldovan, 2020) evaluate the correctness of these links, rather than whether systems can use the connected information to answer questions. Quoref (Dasigi et al., 2019) evaluates reference resolution through question answering, but within a single passage.

To study how well systems combine information across implicitly linked documents, we introduce XRefQA (Cross-Reference Question Answering), a diagnostic benchmark of question answering over small document collections. In each collection, the information needed to answer a question is present but distributed across documents, so identifying and combining it requires resolving the implicit references that connect the evidence. On XRefQA, standard RAG answers only 2% of questions correctly with Claude Sonnet 4.6, and switching to the stronger Opus 4.6 raises this to just 7%.

We propose *context propagation*, an algorithm that enriches individual documents with contextual annotations that make implicit information explicit and easier to interpret (Fig. 1). For each target document, we retrieve supporting documents and use them as evidence to resolve implicit references, identify unnamed entities, and expand underspecified mentions. The resulting annotations are stored alongside the original text, augmenting it with additional context and making each document a richer source of evidence for subsequent rounds. As we iterate over documents and rounds, knowledge about entities, relationships, and events accumulates and propagates across the collection. The result is an increasingly rich and explicit representation of project knowledge, supporting more accurate downstream reasoning while reducing the need to reconstruct missing context at inference time.

On XRefQA, two rounds of context propagation enable the weaker Claude Sonnet 4.6 to achieve 80% accuracy with RAG over enriched documents. We further show experimentally that the benefits of context propagation extend beyond implicit-reference resolution. On the multi-hop question-answering benchmarks, context propagation makes it easier to answer questions whose evidence is scattered across the collection, since a retrieved document's annotations can supply facts from documents that were not themselves retrieved. On Summary of a Haystack (SummHay) (Laban et al., 2024), linking each fact to the other documents that state it improves both fact discovery and source attribution. Together, these results suggest that making cross-document connections explicit in the corpus helps downstream models retrieve, combine, and attribute distributed evidence.

2 Related Work

Reference Resolution and Document Contextualization. Context propagation addresses implicit references whose interpretation requires information from other documents, often because authors assume their readers share background knowledge. Prior work on cross-document coreference and

entity disambiguation studies one part of this problem: determining when mentions refer to the same underlying entity or event (Bagga & Baldwin, 1998; Bunescu & Pasca, 2006; Lee et al., 2012; Cattan et al., 2021). A related line of work makes text more self-contained for downstream tasks, including retrieval and question answering. Decontextualization rewrites context-dependent sentences (Choi et al., 2021), Dense X Retrieval constructs self-contained propositions (Chen et al., 2024), and Contextual Retrieval adds explanatory context from the parent document to chunks before indexing (Anthropic, 2024). These approaches draw context from the source passage or parent document; resolving implicit cross-document references additionally requires recovering context from elsewhere in the collection. Contextual Document Embeddings (Morris & Rush, 2025) incorporate broader corpus context into retrieval representations, but keep that context implicit in embeddings rather than exposing it as evidence to the downstream solver. Context propagation instead retrieves evidence from related documents and records it as source-grounded annotations alongside the original text, making the recovered context available to both retrieval and downstream reasoning.

Cross-Document Knowledge Integration for RAG. Beyond resolving individual references, answering a question may require connecting evidence distributed across several documents. IRCoT (Trivedi et al., 2023) interleaves retrieval and reasoning to connect this evidence at query time. Other methods build reusable representations of this evidence in advance, including RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), HippoRAG (Gutierrez et al., 2024), HippoRAG 2 (Gutierrez et al., 2025), and SiReRAG (Zhang et al., 2025). IndexRAG (Bao & Shi, 2026) is particularly close to our approach: it generates bridging facts from information distributed across documents at indexing time and indexes them as separate retrieval units. The central distinction is how the integrated knowledge is stored: these preprocessing approaches make distributed evidence reusable through summaries (Sarthi et al., 2024), graph indexes (Gutierrez et al., 2025), or separately indexed facts (Bao & Shi, 2026). Context propagation instead incorporates the recovered evidence into the original documents as source-cited annotations, keeping each document as the primary unit. Retrieving a document therefore brings its original content, added context, and source citations to the solver together. These enriched documents can also guide retrieval and provide evidence for later annotation rounds, allowing context to propagate through intermediate documents. A retrieved document can thus supply a missing reasoning hop through its annotations, even when the supporting source is not retrieved at question time.

Benchmarks for Cross-Document Reasoning and Attribution. Existing benchmarks evaluate reference resolution and evidence integration in different settings. Quoref (Dasigi et al., 2019) evaluates question answering that requires resolving references within a supplied passage, while HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022) require combining evidence across passages whose connecting entities are generally named explicitly. NoLiMa (Modarressi et al., 2025), BABILong (Kuratov et al., 2024), and Oolong (Bertsch et al., 2025) broaden long-context evaluation beyond literal retrieval by testing non-obvious associations, reasoning, and aggregation over distributed information. Our diagnostic targets a narrower intersection of reference resolution and cross-document reasoning: models must resolve implicit cross-document references to identify and combine the evidence needed to answer a question.

3 XREFQA: A BENCHMARK FOR RESOLVING IMPLICIT REFERENCES

To measure how well systems handle *implicit cross-document references*, we build XRefQA, a benchmark whose questions require resolving these references to identify and combine evidence from multiple documents. To this end, we construct small synthetic document collections in which the necessary information is present but distributed across documents. The benchmark consists of 20 mini-collections, each built around a shared setting such as a company, marketplace, or project. Each collection contains 20-25 documents totaling roughly 15K tokens. Within each collection, we generate five related entities of the same type and create one question per entity. Each question asks for the current value of an entity attribute, requiring the system to combine information about updates, exceptions, or changes in entity relationships across documents. This yields 100 questions overall, including 50 questions across ten mini-collections in the test set.

To produce each question, we first generated an explicit list of reasoning hops, capturing both the facts needed to answer it and the implicit links connecting them across documents. We then distributed this information across project-style documents, using underspecified mentions such

as "the stylist" or "the catalog" rather than always repeating full entity names. Each document also included plausible but irrelevant surrounding context. Related entities within the same setting introduced further distractors through overlapping roles, terminology, projects, and events.

For example, one question asks: "How many styles does Harlow & Finch currently carry on the Tessara marketplace?" Answering it requires connecting the following facts across documents:

- Harlow & Finch initially carries 185 styles on the marketplace;
- Ren Jørgensen is the lead stylist at Harlow & Finch;
- Ren's latest curation adds 15 new styles to the catalog;
- Ren sets the design direction for items in the Foundry's production queue;
- The Foundry removes 8 underperforming styles from sale on the marketplace.

The base count alone is insufficient. The two updates appear in separate documents with underspecified mentions that name "Ren" and "the Foundry" without naming Harlow & Finch or Tessara directly. The model must therefore first resolve these references to recognize that both updates apply to Harlow & Finch's catalog, and then combine them with the initial count to obtain 185 + 15 - 8 = 192 styles.

We constructed the dataset using scripted skills for a Claude agent. Each collection was manually inspected to verify that its questions had well-defined answers supported by the documents (see Appendix B for generation, examples and quality-control details).

4 Context Propagation via Iterative Document Annotation

We now present *context propagation*, our method for interpreting and connecting evidence across documents. It enriches each document with explicit, source-grounded annotations before downstream retrieval or question answering. The central goal is to surface information that is typically implicit in real-world documents, including unresolved references, underspecified entity mentions, vague temporal expressions, and latent cross-document relationships. Context propagation supplies this context alongside the original text, reducing a QA model's need to reconstruct it at inference time.

Our method proceeds iteratively over a document collection, updating each document's annotation across multiple rounds. For a given target document, we extract named entities from its raw text in the first round and from its annotated version in subsequent rounds. We define named entities as non-generic identifiers that distinguish specific people, organizations, roles, products, projects, events, or other uniquely identifiable elements (the extraction prompt is given in Appendix C). We then retrieve supporting documents from the rest of the collection by combining semantic similarity with lexical matching. Semantic similarity captures broader contextual alignment, while lexical matching uses the extracted entity names to find documents containing those names. These retrieved documents provide the source material for the subsequent annotation step. The approach is illustrated in Fig. 2.

Using the target document and these supporting sources, the annotation model is instructed to resolve coreferences by linking mentions such as pronouns, abbreviated names, and generic descriptions to specific entities. It is also asked to use the available evidence to anchor relative temporal expressions to concrete dates or periods and make implicit relationships between entities, events, and facts explicit. All annotations must be specific and verifiable, with citations identifying whether claims originate from the target document or supporting sources. Conservative linking rules require cross-document connections to be clearly justified and each step in a multi-hop chain to be independently grounded (see Fig. 3). The resulting bracketed annotations are inserted immediately after relevant terms in the target document (see Appendix D.1 for the prompt and examples). These enriched documents then become available as sources for subsequent annotation rounds, allowing context discovered in one part of the collection to inform the interpretation of other documents.

In each subsequent round, the model receives the raw document, its previous annotation, and newly retrieved annotated documents from the rest of the collection. It uses these inputs to generate a fresh annotation rather than incrementally editing the previous version. Earlier annotations are treated as candidate evidence rather than ground truth, so their content may be preserved, revised, merged, or discarded based on the evidence in the source documents. Before reusing a claim, the model verifies that its cited source supports it, helping limit error accumulation. Over successive rounds, these

```
# Inputs: raw_documents, num_rounds
db = DocumentDatabase(raw_documents)
# Initially, each document's annotation is set to its raw text.

for round_id in range(num_rounds):
    for document_id, raw_document in db.raw_documents.items():
        previous_annotation = db.get_annotation(document_id)
        entities = extract_named_entities(previous_annotation)
        input_documents = hybrid_retrieve(
            semantic_query=previous_annotation,
            keyword_query=entities,
            corpus=db.exclude(document_id=document_id),
        )
        new_annotation = annotate(
            input_documents=input_documents,
            previous_annotation=previous_annotation,
            raw_target_document=raw_document,
        )
        db.update_annotation(document_id, new_annotation)
```

Figure 2: Pseudocode for context propagation.

checks and revisions refine annotations and make them more specific, better grounded, and more coherently organized. Vague statements are replaced with precise, evidence-backed claims, while related facts may be consolidated or separated.

After several rounds of annotation, downstream retrieval and question answering operate over enriched documents rather than the original corpus. The annotations in these documents make entity links explicit, resolve references, anchor temporal expressions, and record grounded cross-document facts. This added context makes relevant evidence easier to retrieve and interpret, reducing the need to reconstruct missing context at inference time. Hence, both retrieval accuracy and answer quality improve in settings with distributed and underspecified evidence. Overall, the method turns fragmented, partially specified text into a more interconnected and interpretable representation of project knowledge.

The term *context propagation* reflects how contextual evidence passes and accumulates between documents over multiple annotation rounds. This perspective is inspired by belief propagation in graphical models, where nodes exchange messages to update their local beliefs. Here, each document has a partial view of the underlying project, while annotations carry contextual evidence between documents. Although any document may, in principle, inform another, considering every possible connection would not scale. We therefore use retrieval to define a dynamic neighborhood for each target document, selecting the sources used to update its annotation. Evidence can therefore reach a target indirectly: information recovered in one document may first clarify an intermediate document, whose annotation then helps interpret the target in a later round.

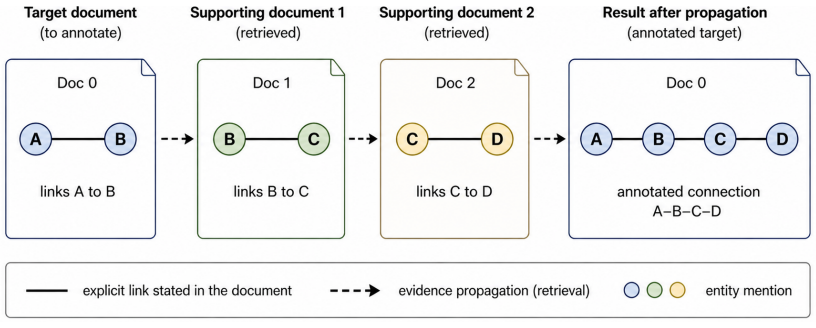


Figure 3: Multi-hop evidence chains across documents. The target document links entity A to entity B; a retrieved supporting document links B to C; another links C to D. By composing these locally grounded links, the annotation exposes the full A–B–C–D connection on the target document.

5 EXPERIMENTS

5.1 QUESTION ANSWERING WITH IMPLICIT CROSS-DOCUMENT REFERENCES

We use XRefQA to evaluate RAG on questions that require resolving implicit cross-document references. We compare RAG over documents enriched with context propagation (CP) against vanilla RAG (Ram et al., 2023), GraphRAG (Edge et al., 2024), HippoRAG 2 (Gutierrez et al., 2025), and Iterative RAG (Trivedi et al., 2023) over the original corpus. We also adapt Contextual Retrieval (Anthropic, 2024), treating the whole collection as the parent document and contextualizing its documents in a single pass. Unless otherwise noted, methods use document-level retrieval with $k = 10$ and Claude Sonnet 4.6 as the QA solver, with reasoning disabled. Iterative RAG uses an agent with one semantic-and-keyword search tool and a budget of 20 turns; we evaluate it both with reasoning disabled and with medium reasoning effort. We score correctness with an LLM judge against the gold answer and report accuracy in Table 1. To test whether a stronger solver alone can overcome the difficulty of the original documents, we run vanilla RAG with Claude Opus 4.6 at medium reasoning effort.

Table 1: XRefQA test accuracy (mean $\pm$ SE, 50 questions). CP and Contextual Retrieval use enriched documents; other methods use the raw corpus. †Medium reasoning effort. ‡Adapted: the whole collection is the parent document.

Setting	Accuracy
Opus 4.6-medium with RAG	
Vanilla RAG†	0.07 ± 0.03
Sonnet 4.6 with RAG	
Vanilla RAG	0.02 ± 0.02
GraphRAG	0.02 ± 0.02
HippoRAG 2	0.00 ± 0.00
Contextual Retrieval‡	0.11 ± 0.03
Iterative RAG	0.03 ± 0.02
Iterative RAG†	0.11 ± 0.03
CP after round 2 (ours)	$\mathbf{0.80 \pm 0.05}$

On the original corpus, vanilla RAG, GraphRAG, HippoRAG 2, and Iterative RAG achieve only 0-3% accuracy. Iterative RAG with medium thinking and Contextual Retrieval each reach 11%, indicating that neither contextual retrieval, graph-structured indexing, nor iterative query-time retrieval overcomes the difficulty of interpreting passages that depend on implicit cross-document references. The difficulty is not merely a matter of solver capability: replacing the Sonnet solver with the stronger Opus model at medium reasoning effort raises vanilla RAG accuracy only to 7%.

Enriching the corpus and then running RAG with the same weaker Sonnet solver lifts accuracy from 2% to 80% after two propagation rounds, far exceeding what the stronger Opus solver achieves over the original documents. Here, enriching the corpus yields a much larger accuracy gain than upgrading the solver.

We also evaluate CP with the open-weight models Kimi K2.6 and Qwen3.6-27B, using each model for both annotation and QA, with reasoning enabled only during annotation. Table 2 reports accuracy across propagation rounds. Sonnet improves from 58% after one round to 80% after two, with no further gain in the third. After three propagation rounds, Kimi's accuracy increases from 3% on the raw corpus to 64%, and Qwen's from 5% to 53%. Unlike Sonnet, both continue to improve from the second to the third round. Thus, CP improves performance across model families, although the benefit of additional rounds varies across models.

Table 2: XRefQA accuracy across CP rounds with RAG. Kimi-K2.6 and Qwen3.6-27B each serve as both annotator and QA solver.

Corpus	Sonnet	Kimi	Qwen
Raw	0.02	0.03	0.05
CP r1	0.58	0.30	0.38
CP r2	0.80	0.53	0.44
CP r3	0.80	0.64	0.53

Additional XRefQA results and analyses, including ablations, retrieval-depth controls, and cost measurements, are provided in Appendix E.1.

5.2 MULTI-HOP QUESTION ANSWERING

Next, we study whether context propagation remains beneficial when cross-document references are explicit. We use three established multi-hop question answering benchmarks, MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018), in which

the entities connecting Wikipedia passages are generally named explicitly. The central challenge is therefore to retrieve and combine evidence distributed across passages rather than to resolve underspecified mentions. We compare RAG over CP-enriched documents with several strong RAG baselines, following the evaluation setting of HippoRAG 2 (Gutierrez et al., 2025, Table 8).

All methods, including CP, use NV-Embed-v2 (Lee et al., 2025) as the retrieval backbone and GPT-4o-mini as the solver. HippoRAG 2 uses an additional LLM call at inference time to filter retrieved triples before question answering. To match HippoRAG 2's number of inference-time LLM calls, we also evaluate CP with reranking (CP*), which uses one additional call to select five passages from the top 20 retrieved passages. To isolate the contribution of reranking, we additionally evaluate NV-Embed-v2 with reranking (NV-Embed-v2*). We report F1 in Table 3 and Recall@5 in Table 5.

Table 3: F1 scores on multi-hop QA benchmarks. Results above the midrule are taken from HippoRAG 2 (Gutierrez et al., 2025, Table 8); results below are our evaluations. *: Includes an additional LLM call for triple filtering (HippoRAG 2) or passage reranking (NV-Embed-v2 and CP).

Method	MuSiQue	2Wiki	HotpotQA	Avg.
NV-Embed-v2	46.0	60.8	71.0	59.3
RAPTOR	39.2	48.4	64.7	50.8
GraphRAG	42.0	61.0	67.6	56.9
LightRAG	9.3	12.1	20.2	13.9
HippoRAG	35.9	67.3	60.0	54.4
HippoRAG 2*	49.3	69.7	71.1	63.4
NV-Embed-v2*	51.4	62.7	74.1	62.7
CP	51.9	71.2	74.5	65.9
CP*	**53.8**	**73.2**	**75.1**	**67.4**

Context propagation remains beneficial even when cross-document references are explicit (Table 3). Without reranking, CP already surpasses HippoRAG 2, the strongest published baseline in this comparison, on all three benchmarks. Adding reranking further improves performance, with CP* achieving the highest F1 on each benchmark. Reranking also improves NV-Embed-v2, but CP* outperforms NV-Embed-v2* on all three benchmarks, indicating that corpus enrichment provides gains beyond reranking alone.

5.3 Understanding Context Propagation Gains

Context propagation gains are not explained by input length alone. Because context propagation augments each passage with annotations, it increases the amount of text shown to the solver. A natural concern is that CP's gains simply reflect having more tokens in context rather than better-organized evidence. To test this explanation, we vary HippoRAG 2's retrieval depth k on MuSiQue (Table 4). At the same k, CP annotations increase the solver's input by roughly 27%, so we increase HippoRAG 2's retrieval depth until it receives a similar or larger number of tokens. Even then, HippoRAG 2 does not close the F1 gap, indicating that CP's advantage comes from better-organized evidence rather than input length alone. We observe the same pattern on XRefQA, presented in Appendix E.1.

Table 4: Comparison of CP and HippoRAG 2 at varying retrieval depths k. Asterisk has same meaning as in Table 3.

Method	k	Tokens	R@k	F1
HippoRAG 2*	5	583	74.2	49.3
HippoRAG 2*	6	700	77.0	50.6
HippoRAG 2*	7	816	79.6	51.0
CP*	5	742	72.6	**53.8**

Decoupling Retrieval and Annotation Gains of CP. CP annotations can improve multi-hop QA through two mechanisms: (i) helping the retriever return more of the gold passages needed to answer a question, or (ii) supplying information from gold passages that were not themselves retrieved. CP achieves higher F1 than HippoRAG 2 despite lower average Recall@5 (Tables 3 and 5), so (i) higher gold-passage recall does not explain its advantage. This suggests that (ii) information supplied through annotations plays an important role. To account for sources cited in annotations, we report *citation-augmented Recall@5* in Table 5, counting a gold passage as covered if it is retrieved directly or cited in an

Table 5: Conventional and citation-augmented Recall@5. HippoRAG 2 results are from (Gutierrez et al., 2025, Table 9).

Method	MuSiQue	2Wiki	HotpotQA	Avg.
HippoRAG 2*	74.2	90.2	95.7	86.7
CP	71.5	84.5	95.4	83.8
CP (citation-aug.)	**78.9**	**97.9**	**98.0**	**91.6**

annotation on a retrieved passage. Including these citations raises CP's average coverage from 83.8% to 91.6%, showing that annotations frequently refer to gold passages outside the retrieved set.

To separate the contributions of retrieval and annotation content, we compare three conditions on MuSiQue (Table 6): (i) retrieval over the original corpus with raw passage text, (ii) retrieval over the CP-annotated corpus while showing the solver only the corresponding raw passages, and (iii) full CP, in which the solver also sees the annotations. The first column in Table 6 shows that changing retrieval alone raises F1 from 46.0 to 47.8; retaining annotations on the same retrieved passages raises it further to 51.9. Therefore, both retrieval and annotation content contribute to the improvement, with annotation content providing the larger gain.

Table 6: F1 for questions resulting in different retrieval content on MuSiQue.

	all questions	all gold facts in retrieved raw docs	annotations supply all missing gold facts	gold facts still missing from retrieved docs
Percentage of questions in CP retrieval	100%	41.3%	9.3%	49.4%
Raw retrieval, raw text	46.0	–	–	–
CP retrieval, raw text	47.8	69.3	41.8	30.9
CP retrieval, annotated text	**51.9**	**70.4**	**64.0**	**34.2**

We further group questions by whether the gold facts required to answer them are available in CP's retrieved passages with and without annotations (remaining columns in Table 6). For 41.3% of questions, all gold facts are already present in the retrieved passages, and annotations provide a modest F1 gain from 69.3 to 70.4. The largest gain occurs for the 9.3% of questions where some gold facts are absent from the retrieved raw passages but annotations supply all the missing facts: F1 rises from 41.8 to 64.0. For the remaining 49.4% of questions, at least one required fact is still missing after annotation, and F1 improves only from 30.9 to 34.2. Thus, annotations are most beneficial when they supply all the missing facts needed to answer a question.

Example of cross-document reference resolution by CP. Figure 4 shows the annotation mechanism on a MuSiQue example. The question asks which label is associated with the performer of "Friend Like That". CP retrieves the gold passage identifying Hawk Nelson as the performer, but the second gold passage, providing the label Fair Trade Services, is absent from its top five. An annotation citing that missing passage supplies the label in the retrieved passage, enabling the solver to answer correctly.

Question: What label is responsible for the performer of "Friend Like That"?

MuSiQue ground-truth reasoning:
Hop 1: "Friend Like That" → performer → **Hawk Nelson** [gold doc: `p07552`]
Hop 2: Hawk Nelson → label → **Fair Trade Services** [gold doc: `p02285`]

CP answer: Fair Trade Services.
HippoRAG 2 answer: Not specified in the text.

Retrieved document (CP):

ID: p07552
TITLE: Friend Like That

"Friend Like That" is the first single from Hawk Nelson's third album, "Hawk Nelson Is My Friend". [Hawk Nelson announced they had signed with Fair Trade Services on December 11, 2012 | ID: p02285]. It was released on Christmas Eve of 2007. This song is on the Digital Praise PC game Guitar Praise. This song was appearing by the compilation album WOW Hits 2009 and his regular versions.

Figure 4: Qualitative MuSiQue example where context propagation completes a missing evidence hop. The first gold passage (`p07552`) is retrieved directly, while the second gold passage (`p02285`) is absent from the top five but is referenced through an annotation attached to the retrieved passage.

We additionally analyze CP's failure modes and annotation quality on MuSiQue in Appendix E.2.

5.4 Multi-document Fact Discovery and Attribution

Next, we evaluate context propagation on Summary of a Haystack (SummHay) (Laban et al., 2024) to test how it extends to a different form of multi-document knowledge use. SummHay constructs collections in which same facts about particular subtopics recur across documents. Given a query about a subtopic, the task is to identify the relevant facts and cite each fact's supporting documents.

We evaluate on complete SummHay benchmark, comprising 10 haystacks, 92 queries, and 621 reference facts. For this task, we adapt the annotation prompt to link each fact in a document to other documents that state it. We follow the benchmark's GPT-4o + Rerank3 configuration. We report coverage for fact recovery and citation for source attribution; the joint metric averages their per-fact product.

Context propagation achieves a joint score of 36.3, compared with 30.8 for the published baseline (Table 7). Both fact recovery and source attribution improve, with coverage increasing from 70.4 to 72.9 and the citation score rising from 41.3 to 47.5.

Table 7: Results on the complete SummHay benchmark. Baseline scores are taken from SummHay (Laban et al., 2024, Table 2); CP scores are from our evaluation.

Method	Coverage	Citation	Joint
GPT-4o + Rerank3	70.4	41.3	30.8
GPT-4o + Rerank3 + CP	**72.9**	**47.5**	**36.3**

6 Conclusion

In this work, we introduced XRefQA, a diagnostic benchmark in which answering questions over small document collections requires resolving implicit cross-document references. Our setting focuses on collections where relevant evidence is present but distributed across documents, with implicit references, evolving entity states, and overlapping roles and events making it difficult to identify and combine. The benchmark highlights the difficulty of using this distributed evidence when its interpretation depends on context from other documents.

We proposed context propagation, an iterative document annotation method that enriches each document using evidence retrieved from related documents in the same collection. The method resolves implicit references, expands underspecified mentions, anchors temporal expressions, and records grounded cross-document relationships directly alongside the original text. Through multiple rounds of annotation, contextual information is progressively propagated across the collection, producing richer and more explicit document representations. This shifts part of the reasoning burden from inference time to corpus preprocessing, making relevant evidence easier for retrieval systems and downstream models to recognize and combine.

Our empirical results show that context propagation substantially improves retrieval-based question answering. On XRefQA, standard RAG fails on almost every question, yet enriching the corpus allows a weaker solver to answer most questions without additional reasoning. Remarkably, enriching the corpus enables this solver to substantially outperform a stronger model using RAG over the original documents. We also find that most gains on this diagnostic emerge within the first few propagation rounds, suggesting that useful context can be recovered through local iterative exchanges between related documents. These benefits extend to multi-hop question answering on MuSiQue, 2WikiMultiHopQA, and HotpotQA, and to multi-document fact discovery and source attribution on SummHay. We discuss the limitations of our approach in Appendix A.

We argue that project understanding should not be treated solely as an inference-time problem. Enriching documents with explicit, source-grounded context can improve downstream retrieval and reasoning while complementing long-context models, retrieval systems, and graph-based RAG methods. Future work should explore more structured representations of propagated annotations, such as navigable project wikis or persistent memory layers, as well as training models to better internalize how information is organized within such knowledge bases. Overall, our results suggest that explicit contextual enrichment is a promising path toward more robust project-level AI systems.

REFERENCES

Anthropic. Introducing Contextual Retrieval, September 2024. URL https://www.anthropic.com/engineering/contextual-retrieval. Accessed: 2026-09-18.

Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th international conference on computational linguistics*, 1998.

Zhenghua Bao and Yi Shi. IndexRAG: Bridging facts for cross-document reasoning at index time. *arXiv preprint arXiv:2603.16415*, 2026.

Amanda Bertsch, Adithya Pratapa, Teruko Mitamura, Graham Neubig, and Matthew R Gormley. Oolong: Evaluating long context reasoning and aggregation capabilities. *arXiv preprint arXiv:2511.02817*, 2025.

Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*, pp. 9–16, 2006.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 5100–5107, 2021.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15159–15177, 2024.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461, 2021.

Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 4545–4552, 2014.

Parag Pravin Dakle and Dan Moldovan. Cerec: A corpus for entity resolution in email conversations. In *Proceedings of the 28th international conference on computational linguistics*, pp. 339–349, 2020.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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. 5925–5932, 2019.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761. Association for Computational Linguistics, 2025.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2024)*, 2024.

Bernal Jimenez Gutierrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.14802.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pp. 6609–6625, 2020.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

Philippe Laban, Alexander Richard Fabbri, Caiming Xiong, and Chien-Sheng Wu. Summary of a haystack: A challenge to long-context llms and rag systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9885–9903, 2024.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. In *International Conference on Learning Representations*, volume 2025, pp. 79310–79333, 2025.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 489–500, 2012.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15339–15353, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *TMLR*, 2025. URL https://openreview.net/forum?id=Cw2xlg0e46.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A Rossi, Seunghyun Yoon, and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching. 2025.

John X Morris and Alexander Rush. Contextual document embeddings. In *International Conference on Learning Representations*, volume 2025, pp. 99737–99749, 2025.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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, EMNLP 2018*, pp. 2369–2380, 2018.

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. Sirerag: Indexing similar and related information for multihop reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 100413–100432, 2025.

A LIMITATIONS

Scale and provenance of XRefQA. XRefQA contains 50 test questions across ten synthetic mini-collections. Privacy and confidentiality concerns restrict access to internal organizational documents and their surrounding context, limiting the public availability of real-world project data containing implicit cross-document references. This motivates our synthetic construction, which allows us to specify and verify the supporting facts and implicit links needed to answer each question. Nevertheless, these controlled collections might not capture the full variability of natural project communication, including references whose supporting context is absent from the collection. Our evaluations on established benchmarks demonstrate broader applicability, but do not directly evaluate implicit-reference resolution in naturally occurring organizational corpora.

Single-sample evaluation and uncertainty. On XRefQA, we generate one answer per question for each solver configuration; standard errors in Table 1 are computed across the 50 test questions rather than repeated decoding runs. Tables 3 and 5 report point estimates without error bars, following the reporting practice in the main result tables of RAPTOR (Sarthi et al., 2024), HippoRAG (Gutierrez et al., 2024) and HippoRAG 2 (Gutierrez et al., 2025). The substantial XRefQA gains recur across model families, supporting the overall finding. Nevertheless, our evaluation does not separately quantify decoding variability, and small differences between configurations should be interpreted cautiously.

Annotation cost. Each propagation round runs the annotator once per document, with retrieved supporting documents in context. Preprocessing cost therefore grows with collection size and the number of rounds, while annotations also increase document length.

Our XRefQA analysis shows that this upfront cost can be offset by cheaper inference (Appendix E.1). Two propagation rounds cost $4.18 for a 24-document collection. Sonnet with RAG over annotated documents costs $0.037 per query and achieves 80.0% accuracy. Opus with medium reasoning effort reading the full raw collection costs $0.118 per query and achieves 75.6% accuracy. Under these measured costs, the upfront expense would be recovered after roughly 50 queries to the same collection.

These savings depend on reusing annotations: infrequently queried or rapidly changing corpora may not provide enough reuse to amortize annotation and reannotation costs. Although our comparisons characterize the cost–accuracy trade-off for the evaluated configurations, they do not establish how best to allocate a fixed total budget between preprocessing and inference across different workloads. We also leave adaptive selection of documents for further annotation and stopping criteria for propagation to future work.

Evaluation scale and online corpus updates. Beyond XRefQA, our multi-hop QA experiments follow HippoRAG 2's evaluation setting and cover approximately 27,600 Wikipedia passages and 3,000 questions across three separate benchmark corpora. We also evaluate on the complete SummHay benchmark. These experiments establish benefits beyond mini-collections, but do not characterize performance and indexing cost at substantially larger scales. Our evaluations also use fixed corpora. We leave extensions to evolving corpora for future work, including updating annotations when source documents are added, revised, or removed and propagating these changes to dependent annotations elsewhere in the collection.

B XREFQA BENCHMARK CREATION AND EXAMPLES

As we describe in Section 3, our benchmark consists of 20 synthetic mini-collections. Each mini-collection is built around a single shared setting, such as a company, marketplace, research project, operational workflow, hospital network, product line, or internal initiative.

Within each mini-collection, we generate five related entities of the same type and instantiate one question per entity. Each question asks for the current state of an entity attribute. The documents are organized into entity-centered groups, with each group containing the evidence needed to answer the corresponding question. Answering the question requires integrating information distributed across multiple documents, rather than locating a single explicit sentence.

We generated the dataset using a set of scripted skills for a Claude agent. The skill specified a multi-stage procedure for constructing document collections with controlled cross-document reference-resolution structure. The generation process proceeded as follows:

1. **Sample a shared collection setting.** Each mini-collection began with a broad business or organizational context, such as a company, marketplace, research project, operational workflow, hospital network, product line, or internal initiative.
2. **Generate five related entities.** Within this setting, we generated five entities of the same type. The entities shared the same broader world and were designed to have related roles, terminology, projects, people, and events.
3. **Create one question per entity.** Each entity was paired with a question asking for the current state of one of its attributes. The question was designed so that the answer could not be obtained from a single explicit sentence.
4. **Specify the required reasoning hops.** For each question, we generated an explicit list of reasoning hops needed to obtain the answer. These hops captured both the factual information required and the implicit links connecting that information across documents. The current state of the target attribute depended on updates, exceptions, revisions, contradictions, or changes in entity relationships.
5. **Generate documents containing the hop information.** The hop information was then distributed across multiple realistic project-style documents, including messages, notes, reports, updates, meeting summaries, and internal communications. Documents were intentionally written with implicit references and underspecified mentions, such as "the stylist," "the catalog," "the Foundry," "the marketplace," "the rollout," or "the handoff," rather than always repeating full entity names.
6. **Introduce realistic distraction.** Distracting information was introduced both within documents and across the mini-collection. Individual documents included plausible but irrelevant surrounding context, while related entities in the same shared setting introduced overlapping roles, terminology, projects, and events. This prevented the task from reducing to keyword matching or simple entity lookup, while preserving a well-defined answer for each question.

Quality control. We manually inspected each mini-collection to verify that every question had a well-defined answer, required multiple reasoning hops, and was supported by the documents. We also checked that distractors did not accidentally reveal the answer or introduce conflicting evidence. When issues were found, we revised the collection with Claude by rewriting documents, removing shortcuts, adjusting distractors, or clarifying the gold reasoning chain.

Dataset splits. We used the training collections to develop the generation pipeline and annotation prompts, and the validation collections to select configurations such as retrieval depth and the number of propagation rounds. The ten test collections, containing 50 questions, were held out until final evaluation.

The five documents below contain the supporting evidence for one question from a mini-collection: *How many styles does Harlow & Finch actively carry in its current lineup on the Tessara marketplace?* Here, "current lineup" excludes discontinued styles whose remaining inventory is being sold through. Answering the question requires the solver to:

1. Identify from Document 1 that Harlow & Finch carries 185 styles;
2. Link Harlow & Finch to Ren Jørgensen via the design attribution in Document 2;
3. Learn from Document 3 that Ren approved 15 new styles in the latest catalog cycle;
4. Link Ren to the Foundry via the production attribution in Document 4;
5. Learn from Document 5 that the Foundry pulled 8 styles from the lineup;
6. Compute $185 + 15 - 8 = 192$.

No single document contains all the evidence needed to compute the answer; the solver must resolve cross-document references to establish which updates apply to the target entity and combine them with the initial count.

Document 1

TIMESTAMP: 20250307-091000
ID: bb779447
Subject: SKU migration discrepancy — catalog sync request

Ticket #4471 — filed by Dariush Kohl, platform operations

A technician submitted this ticket after identifying a discrepancy during the quarterly platform migration audit. The alert was raised while running automated cross-checks between the vendor-facing dashboard and the internal inventory reconciliation ledger. Two storefronts were initially queued for manual review, but only one turned out to have a genuine mismatch between the dashboard figure and the ledger — the second was ruled out as a false positive resulting from a timezone offset in the reporting window. The storefront that required further investigation has been on the platform since the original onboarding phase and has maintained a clean audit record in prior cycles.
Harlow & Finch was identified as the storefront exhibiting the actual discrepancy. When the audit script ran, the vendor dashboard was still serving a stale snapshot from before the most recent batch correction had been applied, which caused the automated alert to fire. The engineering team traced the root cause to a caching layer that retained outdated data for roughly six hours after the correction went through. They have confirmed that the issue was an infrastructure artifact with no impact on the vendor's listing integrity or customer-facing search results. A separate service ticket addressing the caching layer's time-to-live settings has been escalated to the backend infrastructure group, and a patch is expected within the next sprint cycle.
Following the cache flush, the current sync shows 185 entries across all product categories for the flagged storefront. The technician is requesting written confirmation that this corrected figure has fully propagated to the downstream analytics pipeline before the end-of-quarter summary is generated. If a manual trigger is needed to force the pipeline refresh, please coordinate with the reporting team and close out this ticket once reconciliation is confirmed.

Document 2

TIMESTAMP: 20250508-111000
ID: 16fe607a
Customer Support Chat — Session ID: 88291

[Ava Moretti — Customer]: I placed an order a few weeks ago and the stitching on one of the collars looks uneven. I am not sure if it is a defect or part of the intentional design. The item came from Harlow & Finch, and I have ordered from that brand before without issues, so this caught me off guard. I tried checking the product page for details on the construction, but the care instructions only cover washing and storage.
[Nabil Essam — Support Agent]: Thank you for reaching out. I can certainly look into this for you. Could you share the order number so I can pull up the specifics? In the meantime, I want to let you know that asymmetric stitching has come up before with certain lines where the creative team deliberately uses a raw-edge finishing technique. It is not always obvious from the product photos whether a particular collar treatment is meant to look that way or signals a production error.
[Ava Moretti — Customer]: The order number is 7704-AQ. I would appreciate any clarity — I do not want to start a return process if it turns out the design is intentional. I also noticed the fabric feels slightly different from what I received in earlier orders, but that might just be a seasonal material rotation.
[Nabil Essam — Support Agent]: I have pulled up your order and cross-referenced the product batch notes. The collar in question is from a line where Ren Jørgensen introduced an unfinished-edge detail as part of a broader aesthetic direction this cycle. The technique can look irregular at first glance, but it is deliberate. Regarding the fabric feel, you are right that material sourcing rotates and the hand of the cloth can shift between production runs. I will flag your observation for the quality review board, but based on what I see in the batch record, both the collar construction and the fabric weight fall within the expected specification range. Would you like me to send you the detailed product spec sheet for your reference?

Document 3

TIMESTAMP: 20250721-172300
ID: bd36307f
Quarterly Catalog Expansion Report — Internal Distribution

Prepared by: Odette Marchetti, Catalog Operations

This report summarizes listing additions approved during the most recent review cycle. Following a cross-functional evaluation of vendor submissions, the editorial board cleared a batch of 15 new product entries for publication on the platform. The approvals span outerwear, knitwear, and accessories, and each listing has passed both the photography standards audit and the pricing compliance check. Fulfillment routing for the new entries has been configured and warehouse allocation is confirmed, and new listings are now live.

The review panel noted that the overall submission quality this cycle was above the rolling average, with fewer rejections for incomplete size-run documentation than in the prior period. Several of the newly approved items feature construction techniques that had not previously appeared in the marketplace catalog, including a bonded-seam method that reduces visible stitching on outerwear panels. The editorial team flagged these listings for a homepage feature rotation once the marketing calendar has an available slot. A separate merchandising memo covering the feature schedule will be circulated by the end of the week.

The curation lead behind this cycle's submissions is Ren Jørgensen, who coordinated the vendor briefings, oversaw the sample review sessions, and signed off on the final selection. Any questions about the rationale behind individual listing approvals or the criteria applied during the review should be directed to that office. The next scheduled catalog review window opens in six weeks, and submission guidelines will be distributed to vendors at least two weeks in advance.

Document 4

TIMESTAMP: 20250819-121200
ID: 1984d5e7
Comment added by Petra Sundqvist — Production Scheduling

Updating this ticket with the latest handoff status. The production unit referred to internally as the Foundry is currently running at near capacity following a surge in sample requests tied to the upcoming preview window. Turnaround on pattern grading has slipped by roughly two days compared to the standard timeline, and the finishing team is prioritizing orders that have a confirmed wholesale commitment. We have moved three pre-production batches to the secondary cutting room to free up table space, but the bottleneck remains in the pressing and quality-check stages where the same two technicians handle all outbound inspections.

I spoke with the logistics coordinator earlier today about whether we could temporarily redistribute inspection duties, but the compliance handbook requires that final sign-off come from a technician who holds the certified assessor credential, and only the two currently assigned staff meet that requirement. We are looking into an accelerated credential programme for two additional team members, but that will not resolve the short-term backlog. The procurement office has also flagged that one of the dye lots for the next batch may need to be re-sourced due to a supplier substitution, which could push the delivery window further.

The design direction for the items in the queue was set by Ren Jørgensen, so any changes to construction specifications or material substitutions on the pending batches will need sign-off from that side before production can proceed. Please route approval requests through the shared review channel rather than direct messages, as the volume of change orders has made it difficult to track individual threads. I will update this ticket again once the pressing backlog clears and we have a revised delivery estimate.

Document 5

TIMESTAMP: 20251012-100000
ID: 6cb90632
Streamlining What We Carry: A Note on Catalog Curation

By Margaux Tsai

Every season brings a fresh conversation about what belongs on the shelves — or, in our case, what belongs on the screen. Behind the scenes, our production partners put real thought into which pieces justify the resources they demand, from sourcing raw materials through final quality inspection. One production arm that has been particularly deliberate about this question lately is the Foundry, where the team recently completed an internal review of sell-through rates, return frequencies, and margin performance for every item they produce. The process took the better part of a quarter and involved input from warehouse staff, fabric suppliers, and wholesale account managers.

What makes this kind of review difficult is that poor performance is rarely obvious. A product can have decent page views, acceptable conversion, and still quietly erode margin once you account for return-and-restock costs, storage fees, and the opportunity cost of the warehouse space it occupies. The

review panel weighted all of these factors before arriving at its recommendations, and it was clear that a handful of listings were consistently below the viability threshold across every metric examined.
The result of that review is that 8 items have been pulled from the current lineup. The affected listings have already been removed from the marketplace, and any remaining inventory has been withdrawn from sale. No new production runs have been scheduled. For customers, the transition should be seamless — the remaining catalog continues unchanged, and the team is already redirecting capacity toward newer pieces that tested well during the pre-season sampling phase.

C Prompt for Named Entity Extraction

We use the named entity extraction prompt below for XRefQA and the multi-hop question-answering benchmarks. In SummHay, we do not use a separate entity extraction step.

Named Entity Extraction Prompt

```
Extract all named entities and their definitions from the report provided below.

**Named entities** are non-generic proper nouns — names that distinguish a specific person, role,
      organization, team, location, product, or other entity from others of its kind. A generic noun
      phrase that describes what something is rather than naming it is not a named entity, even if it
      refers to a specific thing in context.

Produce the output inside `<output>` tags.
Return each entity and its definition inside `<entity>` tags.
If no named entities are found, return an empty list.

**Format:**

<output>
<entity>
[Entity 1]
Definition: [Definition of Entity 1, using only information present in the report]
</entity>
...
</output>
```

D Annotation Prompts

We use different annotation prompts for XRefQA, the multi-hop QA benchmarks, and SummHay. XRefQA requires resolving implicit references, grounding temporal expressions, making cross-document relationships explicit, and verifying propagated facts against their sources. We therefore use a relatively detailed prompt with the stronger Sonnet 4.6 annotator. For multi-hop QA, the cross-passage entities are generally explicit and we use the weaker gpt-4o-mini as the annotator. We found the full XRefQA prompt unnecessarily complex for this setting and difficult for gpt-4o-mini to follow reliably, so we use a simplified prompt focused on propagating relevant, source-grounded facts across passages. SummHay requires a different form of enrichment: rather than resolving references or reasoning chains, the annotation links facts in each document to other documents that state the same facts, supporting both fact discovery and source attribution. We therefore use a separate task-specific prompt with GPT-4o.

D.1 Annotation prompt for XRefQA

Annotation Prompt for Round 1

```
## Task

You are given:

- A **Background**.
- A set of **Input Reports**.
- A **Target Report**.
```

```
These reports are noisy fragments from different channels observed over time, each revealing only
      partial information about the current state of the underlying world: named entities such as
      people, organizations, projects, and their relationships.

Annotate the Target Report by inserting inline brackets after relevant terms, leaving the original text
      verbatim. Each annotation must include at least one of the following (and may combine several):

- (a) resolving an ambiguity in the Target Report (an unresolved coreference, a relative time reference,
      or an implicit within-report link); or
- (b) adding a concrete cross-report entity relationship or fact drawn from an Input Report that links
      an entity in the Target Report to other named entities or specific events.

**"Concrete"** means a specific named entity, dated event, numeric value, or quoted action — not a
      characterization or general summary.

## What to Annotate

Work in three steps. Step 0 builds understanding. Steps 1 and 2 produce annotations.

### Step 0 — Read and resolve all reports

The goal when reading reports is to extract the maximum factual content from each one. Because the
      reports are noisy, first build a clear understanding of each Input Report by resolving its own
      coreferences, timestamps, and entity mentions, then reveal implicit within-report links: make
      explicit any entity-to-entity or entity-to-fact connections that are present in the report but
      require inference to see. Only then can you reliably identify which facts from Input Reports apply
      to entities in the Target Report.

### Step 1 — Annotate references inside the Target Report

For each named entity in the Target Report:

- **Coreferences** — resolve pronouns (he, she, they), short names, and descriptions ("the company,"
      "the vendor," "the agreement") to the specific entity they refer to. Annotate at the point where
      the reference appears, even if the referent is named elsewhere in the same report — the reader
      should not have to scan forward or backward to identify it.
- **Timeline references** — anchor relative time expressions ("last quarter," "recently," "the previous
      report") to specific dates or periods using the report's timestamp and context from other reports.
- **Implicit within-report links** — make explicit any entity-to-entity or entity-to-fact connections
      that are present in the Target Report but require inference to see.

### Step 2 — Annotate cross-report context from Input Reports

For each named entity A in the Target Report, surface concrete connections to other named entities or
      specific facts found in the Input Reports (Notation below: A is always an entity from the Target
      Report; B, C, D... may be entities or specific facts).

- **Cross-report connections** — identify which other named entities or facts the target entity is
      connected to across the Input Reports, and in what capacity (role, relationship, or transaction).
      Chains may extend across multiple reports
    - A is linked to B in an Input Report → annotate A-B.
    - A-B is in the Target Report; B-C appears in an Input Report → annotate A-B-C.
    - A-B is in the Target Report; B-C is in one Input Report; C-D is in another → annotate A-B-C-D.
    - ...and so on, as long as each new link adds information relevant to understanding A.

  To extract richer chains, use the coreferences and timeline references you resolved during the reading
      phase in Step 0.

- **Specific, time-anchored facts** — annotations must name concrete entities, numeric values, dated
      events, or quoted actions. Annotate time-anchored concrete facts rather than summaries or
      paraphrases of what the source "says generically about" the entity, in order to track how the
      underlying world state evolves over time.
    - "approved 12 engineering hires in Q3 2024" rather than "expanded the team."
    - "served as CFO at Acme from 2019-2022" rather than "previously held a finance role at Acme."

## What NOT to Annotate

### The test: does the source uniquely identify the entity or fact?

Only annotate when the source text uniquely identifies the entity or fact. The identification must be
      grounded in the text — named entities and verifiable facts such as roles, dates, affiliations, and
      identifiers. What matters is whether the source provides enough verifiable detail to pin down one
      entity or fact — enough that a reader could check the match against the source and confirm it,
      without relying on assumptions not present in the text. Note that resolving a source's own
      internal references — its pronouns, possessives, and definite descriptions — is part of reading
      what the source says, not an assumption beyond it; the evidence base includes everything the
      source states, substantiates, or entails through the facts it provides, not just what it names
      explicitly.
```

```
Before adding any cross-report annotation: state each link in the chain as a single proposition, name
      the specific elements in each source that pin that link down, and check that each source provides
      them. A chain that spans multiple reports is safe only if every step is pinned down by its
      respective source. If any single step requires you to close a gap the source left open, do not
      annotate. If every step is grounded, annotate — even if the full chain spans several reports and
      reasoning steps.

## Patterns that require extra care

The following patterns are not forbidden, but frequently leave the identification open. They are flags
      to apply the test above with extra care.

- **Partial identifier** (first name, initials, nickname): "Sarah confirmed the launch" cannot be linked
      to "Sarah Chen, Director of Product at Northwind" unless the surrounding context independently
      establishes Northwind, the launch, or enough detail that narrows to one Sarah.
- **Role or generic descriptor** ("the CTO," "the vendor"): "the CTO approved the timeline" cannot be
      linked to "Lin Wei, CTO at Helios Systems" unless the surrounding context independently
      establishes Helios or enough detail that narrows to one CTO.
- **Shared topic or domain** ("AI," "data privacy"): never sufficient on its own. Two reports mentioning
      the same broad subject are not factually connected by that alone. Require a date, party,
      identifier, or other anchor.
- **Similar but unconfirmed entities** ("the data privacy agreement" vs. "the GDPR compliance
      contract"): a single shared attribute is not sufficient — different entities routinely share one
      date, one party, or one topic. Safe only when multiple independent details corroborate and narrow
      to one specific occurrence.
- **Generic events or processes** ("the quarterly review," "the launch"): the same kind of event recurs
      regularly. A single shared attribute is not sufficient; the combination of anchors must narrow to
      one specific occurrence.

Note that an exact string match is not by itself sufficient — two entities can share a name — and a
      partial or paraphrased reference is not by itself disqualifying. The string is not what pins the
      identification down; the surrounding context is.

## Format

Insert annotations inline in square brackets immediately after the term they annotate, with source
      information. Do not modify, paraphrase, reorder, or remove any text from the Target Report —
      annotations are additions only, and the original term remains unchanged immediately before each
      bracket. For references resolved within the same report, use `this report` as the source. Example:

`entity [annotation | ID: xxx]`

If there are multiple independent annotations, use separate brackets:

`entity [annotation-1 | this report] [annotation-1 | Source-ID-1] [annotation-2 | Source-ID-2]`

If multiple annotations combine in a sensible and rich way, combine them in a single bracket:

`entity [annotation-combined | this report, Source-ID-1, Source-ID-2]`

When you cite a fact, cite the source ID of the report that originally contains the fact.

## Example

Given Input Reports: Public release (ID: 909a4eca), Email (ID: 0d71493c), Slack announcement (ID:
      1e3a4e7a), Report (ID: 909a4e12).

Given Target Report:

> TIMESTAMP: 20221104-121100
> ID: 134b4a91
> Report: Project Update
>
> The project results were announced by Yoshua Dean. ...
>
> ...
>
> Multiple vendors such as AcmeZ and DirAI were involved. ...
>
> ...
>
> He told the vendors that the agreement would continue. ...
>
> ...
>
> Last year's report lists twenty engineers.

Annotated Target Report:
```

```
> TIMESTAMP: 20221104-121100
> ID: 134b4a91
> Report: Project Update
>
> The project results were announced by Yoshua Dean [Yoshua Dean is the project manager of Moonshot
      Project, at Apex Corp. He works with Mark Daniel on the same project | ID: 0d71493c, ID: 909a4eca,
      ID: 1e3a4e7a] [Yoshua Dean approved 12 engineering hires in Q3 2022 | ID: 909a4e12]. ...
>
> ...
>
> Multiple vendors such as AcmeZ and DirAI were involved. ...
>
> ...
>
> Last year's report lists twenty engineers [20 engineers as of end of 2021, given this report is dated
      Nov 2022 | this report].
>
> ...
>
> He [Yoshua Dean | this report] told the vendors [AcmeZ and DirAI | this report] that the agreement
      would continue. ...

Notes on this example:

- **Step 0**: Resolve each Input Report's internal references before annotating. For example, if an
      Input Report uses "he" or "the project" internally, resolve those first to establish the facts
      that report contains.
- **Step 1**: "He" → Yoshua Dean, "the vendors" → AcmeZ and DirAI, and "Last year's report" → 2021
      (given this report is dated Nov 2022). All three are resolved using `this report`. Specific
      verifiable facts rather than general descriptions.
- **Step 2 (combined)**: facts about Yoshua Dean from the Email, Public release, and Slack announcement
      are merged into a single richer bracket because they describe the same role, employer, and project.
- **Step 2 (separate)**: the Q3 2022 hiring fact from the Report is kept in its own bracket because it
      is a distinct event rather than part of the role/employer description. Specific verifiable facts
      rather than general descriptions.

## Output

Produce the annotated Target Report inside `<output></output>` tags. Includes TIMESTAMP, ID fields, plus
      the final annotated Target Report.

------

## Background Summary

Tessara operates an online fashion marketplace connecting independent clothing brands with consumers.
      The platform manages product listings, coordinates fulfillment logistics, and facilitates
      partnerships between brands and brick-and-mortar retailers. Each brand on the platform maintains
      its own creative studio, sources its own materials, and employs a dedicated team spanning design,
      buying, merchandising, and operations.

------

## Input Reports

{input_reports}

## Raw Target Report

{target_report}
```

Annotated Document 1 after Round 1

TIMESTAMP: 20250508-111000
ID: bb779447
Subject: SKU migration discrepancy — catalog sync request

Ticket #4471 — filed by Dariush Kohl, platform operations

A technician [Dariush Kohl, platform operations — the ticket filer identified in the header of this report | this report] submitted this ticket after identifying a discrepancy during the quarterly platform migration audit. The alert was raised while running automated cross-checks between the vendor-facing dashboard and the internal inventory reconciliation ledger. Two storefronts were initially queued for manual review, but only one turned out to have a genuine mismatch between the dashboard figure and the ledger — the

second was ruled out as a false positive resulting from a timezone offset in the reporting window. The storefront that required further investigation has been on the platform since the original onboarding phase and has maintained a clean audit record in prior cycles.
Harlow & Finch [as of March 7, 2025, a customer (Ava Moretti, order #7704-AQ) reported an uneven collar on a Harlow & Finch item; support agent Nabil Essam confirmed the unfinished-edge detail was intentional, introduced by lead stylist Ren Jørgensen as part of a broader aesthetic shift that cycle | ID: 16fe607a] [Ren Jørgensen, identified as Harlow & Finch's lead stylist in ID: 16fe607a, is also identified in an Oct 12, 2025 production scheduling note as the lead stylist setting design direction for all items in the Foundry production unit, which handles manufacturing exclusively for a single vendor storefront on the platform — indicating the Foundry is Harlow & Finch's exclusive manufacturing unit; as of that date the Foundry was running near capacity with a ∼two-day slip in pattern grading turnaround and a bottleneck at the pressing and quality-check stages | ID: 1984d5e7] was identified as the storefront exhibiting the actual discrepancy. When the audit script ran, the vendor dashboard was still serving a stale snapshot from before the most recent batch correction had been applied, which caused the automated alert to fire. The engineering team traced the root cause to a caching layer that retained outdated data for roughly six hours after the correction went through. They have confirmed that the issue was an infrastructure artifact with no impact on the vendor's listing integrity or customer-facing search results. A separate service ticket addressing the caching layer's time-to-live settings has been escalated to the backend infrastructure group, and a patch is expected within the next sprint cycle [i.e., after May 8, 2025, the date of this report | this report].
Following the cache flush, the current sync shows 185 entries across all product categories for the flagged storefront [for cross-tier reference: as of Feb 24, 2025, a separate vendor confirmed 195 products after a similar feed-validation audit (Ticket #5038); Sable & Thread showed 175 as a training-reference figure in a platform onboarding guide; Maison Lierre showed 210 in the Q1 benchmark review — all distinct storefronts from Harlow & Finch | ID: 658be19f, ID: 5866985c, ID: 5017f1a3]. The technician [Dariush Kohl | this report] is requesting written confirmation that this corrected figure has fully propagated to the downstream analytics pipeline before the end-of-quarter summary is generated. If a manual trigger is needed to force the pipeline refresh, please coordinate with the reporting team and close out this ticket once reconciliation is confirmed.

Annotated Document 1 after Round 2

TIMESTAMP: 20250508-111000
ID: bb779447
Subject: SKU migration discrepancy — catalog sync request

Ticket #4471 — filed by Dariush Kohl, platform operations

A technician [Dariush Kohl, platform operations — the ticket filer identified in the header of this report | this report] submitted this ticket after identifying a discrepancy during the quarterly platform migration audit. The alert was raised while running automated cross-checks between the vendor-facing dashboard and the internal inventory reconciliation ledger. Two storefronts were initially queued for manual review, but only one turned out to have a genuine mismatch between the dashboard figure and the ledger — the second was ruled out as a false positive resulting from a timezone offset in the reporting window. The storefront that required further investigation has been on the platform since the original onboarding phase and has maintained a clean audit record in prior cycles.
Harlow & Finch [on March 7, 2025, customer Ava Moretti (order #7704-AQ) reported an uneven collar; support agent Nabil Essam confirmed the unfinished-edge detail was intentional, introduced by lead stylist Ren Jørgensen as part of a broader aesthetic shift | ID: 16fe607a] [Ren Jørgensen is Harlow & Finch's lead stylist; as of August 2025 she served as curation lead for 15 new listings approved across outerwear, knitwear, and accessories — including a bonded-seam method flagged for homepage feature rotation — and signed off on the final batch | ID: bd36307f] [the Foundry production unit handles manufacturing exclusively for Harlow & Finch's storefront; as of October 12, 2025, Ren Jørgensen set design direction for all items in its queue (any construction-specification or material-substitution changes requiring her sign-off), the unit was running near capacity following a surge in sample requests, with pattern grading slipped ∼2 days and a bottleneck at pressing and quality-check stages handled by only two certified-assessor-credentialed technicians | ID: 1984d5e7, ID: 16fe607a] [in July 2025, the Foundry completed an internal review of sell-through rates, return frequencies, and margin performance resulting in 8 items being pulled from the storefront's catalog | ID: 6cb90632] was identified as the storefront exhibiting the actual discrepancy. When the audit script ran, the vendor dashboard was still serving a stale snapshot from before the most recent batch correction had been applied, which caused

the automated alert to fire. The engineering team traced the root cause to a caching layer that retained outdated data for roughly six hours after the correction went through. They have confirmed that the issue was an infrastructure artifact with no impact on the vendor's listing integrity or customer-facing search results. A separate service ticket addressing the caching layer's time-to-live settings has been escalated to the backend infrastructure group, and a patch is expected within the next sprint cycle [i.e., after May 8, 2025, this report's date | this report].
Following the cache flush, the current sync shows 185 entries across all product categories for the flagged storefront [for cross-tier context: as of February 24, 2025, a separate vendor confirmed 195 products across all listed categories after a feed-validation audit (Ticket #5038, filed by Daan Kowalski); Sable & Thread showed 175 as of the April 10, 2025 onboarding guide; Maison Lierre showed 210 against a projected range of 190–200 in the Q1 2025 catalog benchmark — all distinct storefronts from Harlow & Finch | ID: 83fd59e3, ID: 5866985c, ID: 5017f1a3]. The technician [Dariush Kohl | this report] is requesting written confirmation that this corrected figure has fully propagated to the downstream analytics pipeline before the end-of-quarter summary is generated. If a manual trigger is needed to force the pipeline refresh, please coordinate with the reporting team and close out this ticket once reconciliation is confirmed.

D.2 Annotation prompt for Multi-Hop QA Benchmarks

Annotation Prompt for Multi-hop QA

```
## Task

You are given Input Reports and a Target Report to be Annotated. All are passages from Wikipedia
      articles.

Your task is to
1. find named entities that appear both in an Input Report and the Target Report to be Annotated.
2. then integrate a concrete fact about that entity from the Input Report to the Target Report to be
      Annotated.

Format:
Copy the Target Report word for word. After the sentence of the matching named entity (person, place,
      organization, work, ...), insert the concrete fact inside a bracket followed by the Input Report's
      ID:
<sentence with the matching entity>. [concrete fact from an Input Report | Input Report's ID: xxx].

Rules:
- Do not change, delete, or reorder any text of the Target Report. Concrete facts
  inside the brackets are the only additions.
- The facts in the brackets must come from Input Reports with the Input Report's ID.
- Facts must be concrete: names, dates, numbers, events. Do not speculate.
- If an entity does not appear in any Input Report, leave it without a bracket.
- If nothing can be annotated, output the Target Report unchanged.

## Example

{example}

## Output

Write the annotated Target Report, including ID and TITLE, inside <output></output> tags.

------

## Input Reports

{input_reports}

------

## Target Report to be Annotated

{target_report}
```

D.3 ANNOTATION PROMPT FOR SUMMHAY BENCHMARK

Annotation Prompt for SummHay

```
## Task

You are given:

- A **Background** describing the setting all the reports come from.
- A set of **Input Reports**, each with an `ID`.
- A **Target Report to be Annotated**, with its own `ID`.

The reports were written independently of one another. They do not refer to each
other, and they are not ordered in time. What they do share is subject matter: the
same facts recur across several reports, because different reports describe the same
situation.

Annotate the Target Report by inserting inline brackets, leaving the original text
verbatim. An annotation records **corroboration**: the Target Report states a fact,
and one or more Input Reports state the same fact. Record it, and cite every Input
Report that states it.

A fact can be:

- **A specific claim** — a number, a duration, a date, a percentage, a named test,
  product, place, organization, or a quoted position — something a reader could
  check.
- **A recurring interaction** — the same question asked, instruction given, or
  concern raised in another report — provided the concrete content matches.

General subject matter is neither: two reports discussing the same topic do not
share a fact unless the concrete content lines up.

## What to Annotate

Work in two steps.

### Step 1 — List the facts in the Target Report

Read the Target Report and list its facts, as defined above. Ignore its general
discussion.

### Step 2 — Match each fact against the Input Reports

- When several Input Reports state the same fact, list all of their IDs in one
  bracket. Do not stop at the first match — the value of the annotation is the
  completeness of the list.
- Paraphrase is expected and acceptable. What must match is the concrete content,
  not the wording.
- Values that differ are different facts. If an Input Report states a different
  value for the same quantity, do not cite it as corroboration.

## What NOT to Annotate

- **Shared topic is never enough.** Require matching concrete content.
- **A generic descriptor is not an entity.** Roles shared by all reports do not
  identify a specific participant.
- **Annotate each fact once**, at its first clear statement in the Target Report.
  Do not annotate later restatements of the same fact.
- **Never cite the Target Report's own ID.**
- **Do not speculate.** If no Input Report states the fact, leave it unannotated.

## Format

Insert annotations inline in square brackets immediately after the sentence or term
they annotate. Do not modify, paraphrase, reorder, or remove any text of the Target
Report — annotations are additions only.

`... original sentence. [short label of the fact | ID: 7, ID: 23, ID: 41]`

- The label is a short reminder of the fact — a few words, never a full sentence.
- Cite sources as the bare `ID` integers of the Input Reports. Use one bracket per
  distinct fact; combine IDs inside a bracket when they all state that fact.

If nothing in the Target Report can be annotated, output it unchanged.

## Example
```

```
Given Input Reports with IDs 12, 30, 58, 77.

Given Target Report:

> ID: 41
>
> The inspection found that the support beams had corroded after roughly ten years
> of exposure. Repairs are estimated at 2 million dollars, and the crossing will
> remain closed during the work. At the hearing, residents asked whether the detour
> would pass through the school zone. The council said the repair estimate could
> still change.

Annotated Target Report:

> ID: 41
>
> The inspection found that the support beams had corroded after roughly ten years
> of exposure. [corrosion after ~10 years | ID: 12, ID: 30, ID: 77] Repairs are
> estimated at 2 million dollars, [2M repair estimate | ID: 58] and the crossing
> will remain closed during the work. At the hearing, residents asked whether the
> detour would pass through the school zone. [school-zone detour question | ID: 30]
> The council said the repair estimate could still change.

Notes on this example:

- The ten-year corrosion is a specific claim stated by three Input Reports; all
  three IDs are in one bracket.
- An Input Report quoting a different repair figure would not be cited for the
  2-million estimate.
- The residents' question is a recurring interaction, cited because Input Report 30
  raises the same question with the same concrete content.
- "The council said the estimate could change" is general discussion; nothing was
  annotated on it.
- Labels are a few words; the claim is not restated.
- Report 41's own ID never appears as a source.

## Output

Produce the annotated Target Report inside `<output></output>` tags, including its
`ID` line.

------

## Background Summary

The reports below are transcripts of separate conversations that all take place
within one recurring scenario: {scenario}

The participants are {participants}. Each conversation involves different people, so
the same role is filled by a different name in each transcript. The conversations
were recorded independently of one another and are not ordered in time.

Specific factual details — a number, a date, a duration, a named test or product —
frequently recur across several transcripts, because the same situation is discussed
by different participants. Each report carries an `ID` used to cite it.

------

## Input Reports

### Input Report 1
ID: {input_id_1}

{input_document_1}

------

### Input Report 2
ID: {input_id_2}

{input_document_2}

------

...

------

### Input Report 19
```

Table 8: XRefQA accuracy under different retrieval depths, propagation rounds, and full-context inputs. k^* matches the number of distinct sources represented in the retrieved annotated documents. All CP annotations are generated by Sonnet 4.6.

Corpus representation	Documents supplied	Sonnet 4.6	Opus 4.6, medium
Raw	10	0.02	0.07
Raw	15	0.09	0.13
Raw	k^*, mean 18.9	0.13	0.38
Raw	20	0.04	0.47
Raw	Full collection	0.29	0.76
CP, round 1	10	0.58	0.69
CP, round 2	10	0.80	0.87
CP, round 3	10	0.80	0.82
CP, round 3	Full collection	0.87	1.00

```
ID: {input_id_19}

{input_document_19}

------

## Target Report to be Annotated

ID: {target_id}

{target_document}
```

E Additional Results and Analysis

E.1 XRefQA

We extend the evaluation in Section 5.1 to examine how context propagation (CP) interacts with evidence access, solver capability, and iterative retrieval. We then isolate its main components and report the complete open-weight annotator–solver comparison. All tables below report accuracy as a proportion.

Evidence access and solver capability. Retrieved annotated documents can contain information from sources outside the retrieved set. To measure this additional coverage, we count, for each question, the union of the ten retrieved documents and all distinct documents cited in their annotations. This union contains 18.9 documents on average, approximately 79% of the collection, with a range of 10–24 documents. We denote the per-question count by k^* and evaluate raw retrieval at this depth, alongside fixed depths of 10, 15, and 20 documents. We also provide the full collection directly to the solver, removing query-time retrieval altogether.

Table 8 reports these comparisons for Sonnet 4.6 and Opus 4.6, together with results across propagation rounds. Both solvers use the same Sonnet-generated annotations; Opus runs at medium reasoning effort. This comparison tests whether CP remains useful with a stronger solver, rather than only compensating for the weaker solver's limitations.

Increasing access to raw documents substantially improves Opus, but does not close the gap to CP. At the matched document count k^*, raw retrieval reaches 13% with Sonnet and 38% with Opus, compared with 80% and 87% after two propagation rounds. Even providing the entire raw collection yields only 29% and 76%, respectively. Thus, although annotations expose information from additional sources, increasing raw retrieval depth alone does not reproduce their benefit in these experiments.

The gains also persist within the stronger solver: Opus improves from 7% to 87% with retrieval. Moreover, enriching the full collection raises Sonnet from 29% to 87% and Opus from 76% to 100%. These full-context comparisons show that enrichment helps even when every original document

Table 9: Effect of one propagation round on single-pass and iterative RAG. The Iterative RAG configuration uses medium reasoning effort.

Inference procedure	Raw corpus	CP, round 1
Single-pass RAG	0.02	0.58
Iterative RAG, medium	0.11	0.61

Table 10: Component ablations on XRefQA. All configurations use two propagation rounds. Each row modifies the full CP pipeline only as described.

Component	Configuration	Accuracy
Reference	Full CP pipeline	0.80
Previous annotation	Do not provide the target document's previous-round annotation to the annotator	0.80
Annotation retrieval	Use only semantic retrieval to select supporting documents for annotation	0.42
Temporal grounding	Remove instructions for resolving and grounding temporal expressions	0.71
QA retrieval	Use only semantic retrieval at question time	0.76
QA retrieval	Use only lexical retrieval at question time	0.76
Citation tags at QA	Remove citation tags from annotations before passing them to the solver	0.78
Source verification	Remove source-verification instructions from the annotation prompt	0.74

is available to the solver. Opus's 100% accuracy also provides a solver-based sanity check of answerability: every evaluated question is answered correctly when the full annotated collection is available. However, additional propagation is not uniformly beneficial. With retrieval, Sonnet remains at 80% in round 3.

Interaction with iterative retrieval. We next test whether iterative query-time retrieval can substitute for corpus enrichment. On the raw corpus, Iterative RAG with Sonnet at medium reasoning effort improves accuracy from 2% to 11% (Table 9), but remains far below CP. After one propagation round, however, single-pass and Iterative RAG perform similarly, at 58% and 61%, respectively. Thus, iterative search provides only a small additional benefit once the documents have been enriched, suggesting that much of the difficulty lies in making the cross-document relationships explicit rather than in performing more query-time retrieval.

Inspection of the raw-corpus Iterative RAG transcripts reveals two recurring failure modes. In one, the agent retrieves a relevant fact but cannot establish that it concerns the entity in the question, and declines to answer. In the other, it commits to an outdated value because its retrieved evidence does not reveal a subsequent update. The agent uses only three to four turns on average and does not reach the 20-turn limit, so these failures are not caused by exhausting that limit.

These observations help explain why using the same retrieval system during annotation and question answering can produce different outcomes. Question-time searches start from the question and the evidence already found. Annotation-time searches instead start from each document's entities and operate across the collection independently of any question. They can therefore expose intermediate relationships that a question-conditioned search may not discover, making those relationships available before the solver begins answering.

Component ablations. Table 10 examines which components of CP are responsible for its gains. Withholding the target document's own previous-round annotation has no effect on accuracy, which remains at 80%. Importantly, the annotator still receives annotated supporting documents from the rest of the collection, so this result suggests that carrying forward the target's own previous annotation is not necessary when propagated context is available through its neighbors. The largest effect comes from annotation-stage retrieval: using only semantic retrieval to select supporting documents reduces accuracy from 80% to 42%. This highlights the importance of entity-based retrieval for exposing the

Table 11: Open-weight annotator–solver combinations on XRefQA. Raw accuracies are reproduced as reported for each configuration.

Annotator	Solver	Raw	Round 1	Round 2	Round 3
Kimi K2.6	Kimi K2.6	0.03	0.30	0.53	0.64
Kimi K2.6	Qwen3-32B	0.03	0.11	0.31	0.40
Kimi K2.6	Qwen3.6-27B	0.05	0.24	0.47	0.51
Qwen3.6-27B	Kimi K2.6	0.03	0.33	0.53	0.56
Qwen3.6-27B	Qwen3-32B	0.03	0.16	0.36	0.44
Qwen3.6-27B	Qwen3.6-27B	0.05	0.38	0.44	0.53

cross-document connections needed during annotation. Removing temporal-grounding instructions produces a smaller drop to 71%.

In contrast, the system is less sensitive to the retrieval mechanism used at question time. Using only semantic or only lexical retrieval yields 76% accuracy in both cases. We also separate the role of citation tags from source verification. Removing citation tags before passing annotations to the solver has little effect, reducing accuracy only to 78%, whereas removing the annotator's source-verification instructions lowers accuracy to 74%. This suggests that the main value of grounding comes from requiring the annotator to verify propagated claims against their sources, rather than from exposing citation tags to the downstream solver.

Transfer across annotators and solvers. The main text reports configurations in which the same open-weight model performs annotation and question answering. Table 11 extends this comparison to all evaluated combinations of two annotators –Kimi K2.6 and Qwen3.6-27B– and three solvers –Kimi K2.6, Qwen3-32B, and Qwen3.6-27B. Reasoning is enabled during annotation and disabled during question answering, and retrieval returns ten documents.

Both annotators improve all three solvers. The benefit also transfers across model families: Kimi-generated annotations raise Qwen3-32B from 3% to 40%, while Qwen-generated annotations raise Kimi from 3% to 56%. Thus, the usefulness of the annotated corpus does not require the annotator and solver to be the same model or belong to the same family. All six configurations continue to improve from round 2 to round 3, unlike the primary Sonnet configuration. The benefit of additional rounds therefore depends on the annotator–solver combination rather than following a universal two-round stopping point.

Document growth and cost amortization. Additional rounds also increase the amount of text stored and supplied to downstream models. In the Sonnet annotation runs, average document length grows from 418 tokens in the raw corpus to 668, 788, and 836 tokens after rounds 1, 2, and 3, respectively. The relative increases become smaller across rounds, approximately 60%, 18%, and 6%, while Sonnet's retrieval-based accuracy remains unchanged between the second and third rounds.

The upfront annotation cost can nevertheless be recovered through repeated use of the enriched corpus. Two rounds cost $4.18 for a 24-document collection. Subsequent Sonnet queries cost $0.037 at 80.0% accuracy, compared with $0.118 at 75.6% accuracy for Opus reading the full raw collection. Under these measured costs, the annotation expense is recovered after roughly 50 queries. Excluding preprocessing, full-context Opus and Iterative RAG with medium reasoning effort cost 3.4 and 21 times as much per correct answer, respectively. These comparisons describe the evaluated configurations; their advantage depends on sufficient reuse of the annotations to amortize preprocessing. Detailed API accounting and runtime measurements are provided in Appendix F.1.

E.2 Multi-hop Question Answering

In this section, we provide additional analyses on MuSiQue, where we investigate failure modes and annotation quality of context propagation.

Failure mode analysis. We first examine CP's remaining failure modes on MuSiQue. The dominant failure mode is still missing evidence: in 65.6% of incorrect answers, a required gold passage

is neither retrieved nor represented by an annotation. In 23.7% of cases, all gold passages are available but the solver nevertheless fails. The annotation-specific errors are less frequent but more revealing. In 6.7% of failures, the required fact is successfully propagated but the solver still answers incorrectly. In another 4.0%, CP reaches the correct missing source but fails to propagate the required fact. Separately, in 1.6% of incorrect answers, the solver takes its wrong answer directly from an annotation. These annotation-related cases distinguish three remaining challenges: using propagated evidence correctly, selecting the right information to propagate, and preventing erroneous annotations from influencing the final answer. The latter two concern the quality of the propagated annotations themselves, motivating a direct audit of their content and provenance.

Table 12: Failure modes among the 506 incorrectly answered MuSiQue questions. The first four rows partition the primary failure modes; the final row separately reports cases in which an annotation directly contributes the wrong answer.

Failure mode	Fraction
Gold passage missing and not covered by an annotation	65.6%
All gold passages retrieved, but solver fails	23.7%
Missing fact supplied by annotation, but solver fails	6.7%
Missing gold passage cited, but required fact not propagated	4.0%
Wrong answer taken from an annotation	1.6%

Annotation quality. We therefore manually inspect 50 propagated annotations on MuSiQue. Most annotations add genuinely new information that is faithfully supported by their cited source: 36 of 50 cases (72%) fall into this category. The most common non-useful outcome is redundancy. In 7 cases, the annotation simply restates information already present in the target passage and therefore adds no new information. We observe one citation error in which the propagated fact is supported by another input passage but attributed to the wrong source. More substantive content errors are less frequent: 2 annotations assert information not supported by any input passage, while 3 are contradicted by the available evidence or attribute a source fact to the wrong entity. Overall, the audit suggests that most propagated annotations contribute correct new information, while the remaining failures arise primarily from redundant propagation and, less frequently, incorrect or improperly sourced content.

Table 13: Manual audit of 50 propagated annotations on MuSiQue.

Annotation outcome	Count
Supported new fact	36
Restates target passage	7
Correct fact, wrong source	1
Unsupported	2
Contradicted	3

F Cost and Runtime Analysis

F.1 XRefQA

We report API cost, output-token usage, and annotation runtime for XRefQA. Costs are computed from the logged API usage of the runs used in the paper.

Annotation cost. Table 14 reports the preprocessing cost of context propagation, averaged over one mini-collection. Each round costs approximately $2.1 per collection. Two rounds, the configuration used for the main XRefQA result, therefore cost approximately $4.18 per collection.

Table 14: Context-propagation cost on XRefQA, averaged over one mini-collection.

Round	Cost / collection	Output tokens / collection
1	$2.094	103,514
2	$2.082	76,760
3	$2.036	67,534
Total	$6.212	247,808

Annotation runtime. Annotation is parallelized across documents within each collection. A round takes approximately 4–5 minutes on a representative mini-collection, so three sequential rounds take approximately 15 minutes. Collections can also be processed in parallel.

Solver cost. Table 15 reports the API cost of the main XRefQA solver configurations. Across these evaluations, solver calls cost $21.45 and produce 527,431 output tokens. RAG over the annotated corpus remains relatively inexpensive at inference time: the round-2 configuration costs $0.037 per query on average, compared with $0.118 for Opus reading the full raw collection.

Table 15: API usage for the main XRefQA solver evaluations. Costs and output-token counts are totals over each configuration; average cost is reported per query.

Model	Thinking	Setting	Cost	Avg. cost	Output tokens
Opus	none	full raw	$3.850	$0.077	40,894
Opus	medium	full raw	$5.900	$0.118	124,933
Sonnet	medium	full raw	$5.050	$0.101	222,440
Sonnet	none	RAG raw	$1.200	$0.024	34,721
Sonnet	none	CP r1 + RAG	$1.650	$0.033	36,277
Sonnet	none	CP r2 + RAG	$1.850	$0.037	33,731
Sonnet	none	CP r3 + RAG	$1.950	$0.039	34,435
Total			$21.450	–	527,431

F.2 Multi-hop QA Benchmarks

We compute API costs from logged usage at `gpt-4o-mini` list prices. Context propagation uses one annotation round per corpus. Named entities are extracted once per passage and reused during annotation retrieval; NV-Embed-v2 embeddings are computed locally and therefore incur no API cost. Across the three benchmarks, entity extraction costs $4.43 and annotation costs $10.26, for a total preprocessing cost of $14.69.

Table 16: Cost of context propagation on the multi-hop QA benchmarks (one annotation round, `gpt-4o-mini`). Time is annotation wall-clock with 20 concurrent requests.

		Entity extraction		Annotation		
Dataset	Passages	Cost	Output tokens	Cost	Output tokens	Time
MuSiQue	11,656	$1.81	2.16M	$4.22	1.87M	86 min
2Wiki	6,119	$0.93	1.09M	$2.20	0.96M	13 min
HotpotQA	9,811	$1.69	2.04M	$3.84	1.76M	40 min
Total	27,586	$4.43	5.30M	$10.26	4.59M	139 min

F.3 SummHay

For SummHay, annotation costs are taken directly from logged API usage. Reader and judge costs are reconstructed from the saved prompts and outputs, tokenized with `o200k_base` and priced at the corresponding list rates: $5/$15 per 1M input/output tokens for the `gpt-4o-2024-05-13` reader and $2.50/$10 for the GPT-4o judge.

Annotation cost. SummHay requires a single propagation round over 10 haystacks containing 1,000 documents in total. The full annotation pass costs $66.99 and produces 1.31M output tokens, corresponding to approximately $0.067 per document. Table 17 breaks this cost down by domain.

Annotation runtime. Annotation is parallelized across documents within each haystack. Annotating a 100-document haystack takes 146–257 seconds (188 seconds on average), including index construction and job start-up. Since SummHay uses only one propagation round, all 10 haystacks take about 31 minutes when processed sequentially.

Table 17: Context-propagation cost on SummHay using one annotation round with GPT-4o. Per-haystack values are averages within each domain.

Domain	Haystacks	Cost / haystack	Out. tokens / haystack	Total cost	Total out. tokens
Conversation	5	$6.878	123,674	$34.388	618,368
News	5	$6.521	137,354	$32.606	686,769
Total	10	$6.699	130,514	$66.994	1,305,137

Solver and judge cost. Table 18 reports evaluation cost for the four retrieval configurations. Each configuration makes one reader call for each of the 92 queries. The resulting summaries are then evaluated with one judge call for each of the 621 insights. From our logs, we see that the complete Rerank3 evaluation for both raw and annotated documents, including solver and judge calls, takes approximately 5 minutes.

Table 18: API usage for SummHay evaluation with a 15K-token retrieval budget.

Retriever	Documents	Calls	Cost	Avg. cost	Output tokens
Rerank3	raw	92	$7.31	$0.080	28,237
Rerank3	CP r1	92	$7.33	$0.080	28,908
Ours (hybrid)	raw	92	$7.31	$0.079	27,832
Ours (hybrid)	CP r1	92	$7.34	$0.080	29,165
Reader calls		368	$29.29	–	114,142
Judge calls		2,484	$7.43	$0.003	40,545