

Beyond Retrieval: Structuring Evolving and Inconsistent External Knowledge with Proposition Relations for RAG

Noriki Nishida¹ Fei Cheng^{2,3} Yuji Matsumoto^{4,5}

¹National Institute of Advanced Industrial Science and Technology (AIST)

²Kyoto University ³NII LLMC ⁴Tohoku University ⁵RIKEN

noriki.nishida@aist.go.jp

Abstract

Real-world knowledge bases evolve over time and accumulate outdated statements, contradictions, and misinformation. LLMs have been reported to be vulnerable to such inconsistent contexts. In this paper, we propose Proposition-Structured RAG (ProStruct-RAG), a framework that continuously structures evolving and inconsistent external knowledge bases using timestamped propositions and inter-proposition relations. ProStruct-RAG represents a document collection as a graph whose nodes are propositions and whose edges capture temporal updates, contradictions, and support relations, retrieves query-relevant knowledge as a relational structure, and uses it for answer generation in RAG. Experiments on multiple open-book temporal QA benchmarks show that ProStruct-RAG outperforms baselines that do not use inter-proposition relations, suggesting that RAG over evolving and inconsistent knowledge benefits from organizing external knowledge before retrieval, rather than relying solely on inference-time reasoning. We release our code and Asteria, a new fictional temporal QA dataset designed to mitigate data contamination.¹

1 Introduction

The internal knowledge of Large Language Models (LLMs) inevitably becomes outdated. To address this problem, continual fine-tuning (Dhingra et al., 2022; Jang et al., 2022; Zheng et al., 2025) and localized parameter editing (De Cao et al., 2021; Mitchell et al., 2022; Xu et al., 2025b; Shi et al., 2025; Wang et al., 2025) have been proposed to update the internal knowledge. However, because these methods repeatedly update model parameters, they are prone to catastrophic forgetting and make it difficult to maintain long-term knowledge consistency (Onoe et al., 2023; Gekhman et al., 2024; Niu et al., 2025). Retrieval-Augmented Generation

(RAG), which retrieves an external knowledge base at inference time to generate evidence-grounded answers, has therefore been widely adopted (Lewis et al., 2020; Izacard et al., 2023; Laitenberger et al., 2025).

However, external knowledge bases are not always static and consistent. They continuously evolve and are often accumulated in an unorganized form, containing temporally outdated statements, mutually conflicting claims, and unreliable content. Recent efforts on long-context modeling, reasoning-enhanced LLMs, and agentic systems aim to let LLMs integrate information and resolve contradictions from such inconsistent contexts at inference time (Li et al., 2024; Du et al., 2025; An et al., 2025). However, LLMs have been reported to be vulnerable to contexts that mix temporal updates, contradictions, and misinformation (Ouyang et al., 2025; Hou et al., 2024).

To address this challenge, prior work has organized external knowledge before retrieval. For example, Ge et al. (2025) showed that temporally organizing multi-session dialogues as timelines improves memory retrieval and reasoning. Li et al. (2025) decomposed documents into *propositions* (Chen et al., 2024; Liu et al., 2025) and dynamically labeled their truthfulness. However, these methods organize knowledge as collections of facts, without explicitly modeling rich relations between them. Although other work structures knowledge through entity relations (Edge et al., 2024; Han et al., 2025; Nishida et al., 2026), temporal updates, contradictions, and support arise between individual facts rather than entities. This distinction is important because temporal validity and misinformation are not properties of isolated propositions alone.

In this paper, we aim to continually organize evolving and inconsistent external knowledge bases through relations between propositions. Our goal is to examine whether part of the burden of in-

¹<https://github.com/norikinishida/kapipe>

tegrating such knowledge can be shifted from inference-time reasoning to pre-retrieval knowledge organization. Our core hypothesis is that explicit inter-proposition relations help LLMs integrate evidence and resolve temporal updates, contradictions, and misinformation. Specifically, we propose Proposition-Structured RAG (**ProStruct-RAG**), which represents external knowledge as a set of timestamped propositions and structures it through inter-proposition relations. First, we extract concise propositions from a document collection and assign each proposition a timestamp based on the publication date of its source document. Next, we identify updates, contradicts, and supports relations between propositions and construct a proposition graph. At inference time, we retrieve a query-specific subgraph relevant to the user question. Finally, we convert the retrieved query-specific subgraph into a structured context, and the QA model generates the final answer.

We evaluate ProStruct-RAG on open-book temporal QA benchmarks, where answers can change with the target time and models must interpret retrieved knowledge under temporal updates and contradictions. Our results show that proposition decomposition alone does not consistently improve performance, while adding temporal information yields substantial gains. Making temporal update, contradiction, and support relations explicit further improves overall performance, surpassing all baselines on average. These results indicate that RAG over evolving, inconsistent knowledge benefits from organizing external knowledge through inter-proposition relations before retrieval, rather than leaving integration and contradiction resolution entirely to inference-time reasoning.

2 Related Work

Long-Context RAG for Knowledge Obsolescence. LLMs store factual knowledge internally, but it inevitably becomes outdated, degrading performance on time-sensitive tasks (Hu et al., 2025; Song et al., 2025; Gekhman et al., 2025; Luu et al., 2022). Continual fine-tuning and parameter editing update such knowledge (Dhingra et al., 2022; Jang et al., 2022; Zheng et al., 2025; De Cao et al., 2021; Mitchell et al., 2022; Xu et al., 2025b; Sun et al., 2025; Shi et al., 2025; Wang et al., 2025), but suffer from catastrophic forgetting and poor temporal or relational consistency (Onoe et al., 2023; Gekhman et al., 2024; Niu et al., 2025). Retrieval-

Augmented Generation (RAG) addresses outdated or incomplete internal knowledge by consulting external sources (Lewis et al., 2020; Izacard et al., 2023; Laitenberger et al., 2025), but errors often stem from missing or noisy evidence (Yang et al., 2024; Leto et al., 2024). While long-context models can outperform standard RAG (Li et al., 2024), simply increasing context length can even degrade reasoning performance (Goldman et al., 2024; Bai et al., 2025; Gupta et al., 2025; Du et al., 2025; An et al., 2025). These findings motivate approaches that structure long-context information rather than relying on unstructured context expansion.

Knowledge Organization for RAG. Prior work shows that RAG systems and LLMs are vulnerable to temporally evolving facts, conflicting evidence, and false information (Hou et al., 2024; Ouyang et al., 2025; Tan et al., 2023). To address temporal change, several benchmarks and methods organize knowledge along timelines (Kasai et al., 2023; Zhu et al., 2025), while others maintain truthfulness labels for each fact (Li et al., 2025) or dynamic representations such as temporal knowledge graphs and time-aware memories (Lu et al., 2024; Ge et al., 2025). However, these phenomena are typically studied in isolation, despite frequently co-occurring in practice. In contrast, this work enables unified evaluation and modeling of their interactions within a single framework.

Proposition-based RAG and Context Enrichment. Related work explores the appropriate unit of knowledge and retrieval granularity, including proposition-level retrieval (Chen et al., 2024), structural expansion of retrieved units (Tao et al., 2025), and fact extraction pipelines tailored for verification (Liu et al., 2025). Other studies revisit retrieval objectives, arguing that semantic similarity alone is insufficient and proposing retrievers optimized for downstream utility (Xu et al., 2025c,a). Several approaches also enrich prompts with supplementary information, such as timestamps (Kasai et al., 2023; Zhu et al., 2025) or hypothetical retrieval scenarios (Lee et al., 2025). While these methods improve retrieval granularity, selection criteria, or contextual richness, they largely treat retrieved knowledge as independent units. Combining proposition-level retrieval (Chen et al., 2024) with timestamp information (Kasai et al., 2023; Zhu et al., 2025) follows this design: time information is attached to individual propositions, while their temporal and logical relationships remain im-

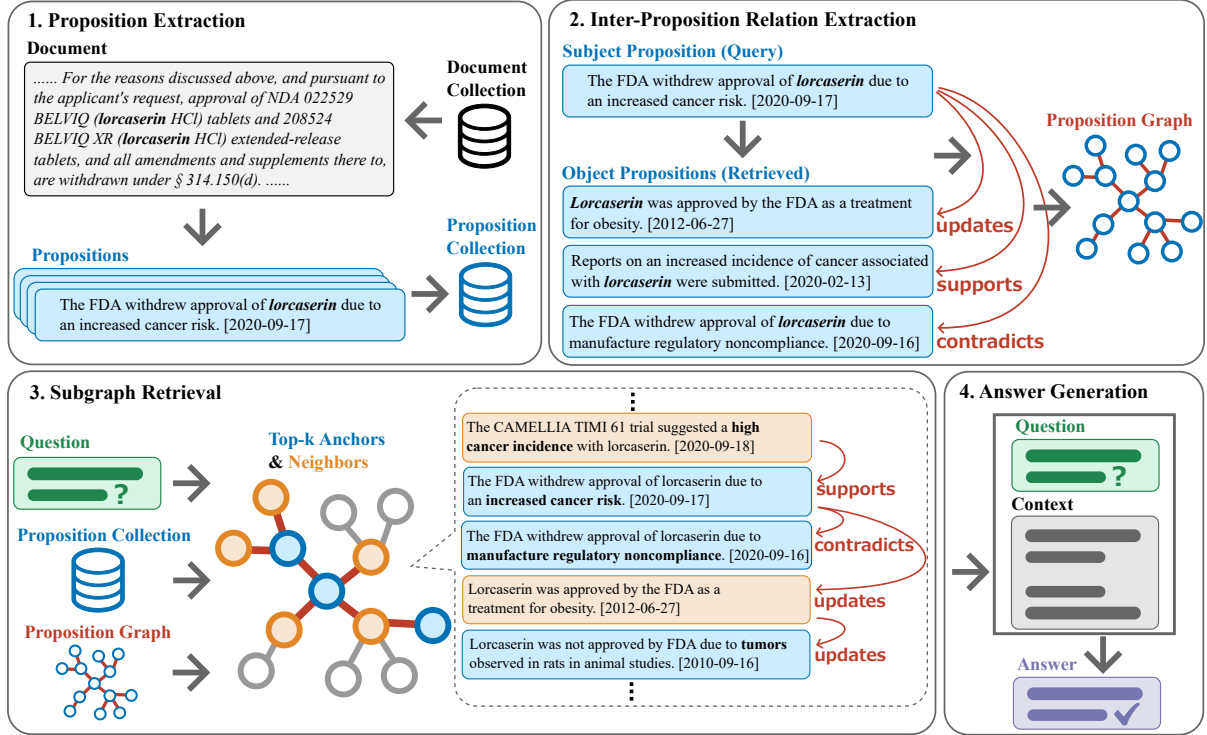


Figure 1: Overview of the proposed ProStruct-RAG framework. ProStruct-RAG is designed to continuously organize evolving and potentially inconsistent external knowledge for retrieval-augmented generation by representing documents as timestamped propositions, explicitly modeling inter-proposition relations, and retrieving compact, relation-aware subgraphs to support robust temporal reasoning.

plicit. In contrast, ProStruct-RAG explicitly organizes propositions through *updates*, *contradicts*, and *supports* relations before retrieval, and uses the resulting structure both for neighborhood-based retrieval expansion and for constructing the context presented to the QA model. Our experiments directly compare these settings to examine whether pre-retrieval relational organization provides benefits beyond proposition decomposition and timestamps alone.

3 Methods

ProStruct-RAG aims to continually organize an evolving and inconsistent external knowledge base into a graph of timestamped propositions and inter-proposition relations. At inference time, it retrieves and presents relevant knowledge as a relational structure, enabling LLMs to better integrate information and resolve temporal updates, contradictions, and misinformation.

Figure 1 shows the framework overview. ProStruct-RAG has four steps: **Proposition Extraction** decomposes documents into concise, self-contained propositions with temporal information; **Inter-Proposition Relation Extraction** identifies

updates, contradictions, and support relations to build a proposition graph; **Subgraph Retrieval** retrieves question-relevant anchor propositions and expands them to a query-specific subgraph with neighboring propositions and relations; and **Answer Generation** linearizes the subgraph into structured textual context for the QA model. The first two steps are performed offline, while retrieval and generation are performed at inference time.

3.1 Proposition Extraction

Definition of propositions. A *proposition* is a concise, self-contained mention of a fact, used as the minimal knowledge unit in ProStruct-RAG. Context-dependent expressions such as pronouns are replaced with their corresponding entities so that each proposition can be interpreted independently. Each proposition is assigned a timestamp in the YYYY-MM-DD format, using the publication date of its source document.² This timestamp is used in downstream steps.

²An alternative is to use the time when the described fact became true, or the period during which it remained true. While more direct, this information is often unavailable in real-world documents and difficult to define for temporally invariant facts. We therefore use source publication dates, which can be obtained stably from document collections.

Relation	Definition
updates	A temporal update relation where the subject partially or fully updates the object. The subject represents a newer state, while the object represents previously valid information that has since been updated.
contradicts	The subject and object make mutually incompatible claims about the same state, so both cannot be true simultaneously.
supports	The subject provides evidence, a reason, or a premise supporting the claim made by the object.
NOREL	Assigned when there is no clear logical relation between the subject and object, including mere paraphrases or redundant information.

Table 1: Definition of inter-proposition relation labels.

Extraction procedure. We use an LLM to extract propositions from the document collection. For each document, the LLM generates multiple concise, self-contained propositions in a single pass, guided by a small set of manually annotated examples. Each proposition inherits the timestamp of its source document. We apply this process to all documents and aggregate the outputs into the *proposition collection* used in subsequent steps. The prompt template used for proposition extraction is provided in Appendix A.

3.2 Inter-Proposition Relation Extraction

Definition of inter-proposition relations. Existing methods often assign factuality labels to propositions independently. In contrast, we represent factuality through inter-proposition relations, modeling whether a proposition updates, contradicts, or supports another proposition. We define inter-proposition relations as directed: the subject (head) is a temporally *newer* proposition, and the object (tail) is from the *same* or an *earlier* time. This constraint captures how newer information updates, contradicts, or supports existing knowledge. Table 1 shows the four relation classes.

Extraction procedure. Classifying all proposition pairs is inefficient, so we first retrieve candidate pairs likely to have relations. For each proposition, treated as the subject, we retrieve the top

20 semantically related propositions with the same or earlier timestamps as object candidates using an off-the-shelf dense retriever.³ This limits subsequent classification to likely relation pairs.

Then, for each candidate pair, an LLM classifies the relation as one of the four classes in Table 1: updates, contradicts, supports, or NOREL. For efficiency, candidate pairs sharing the same subject are grouped into a single prompt and classified in batch. As in proposition extraction, we use few-shot examples covering all relation classes.

To improve relation quality, we further refine pairs initially labeled as relations, i.e., labels other than NOREL. A stronger LLM re-evaluates each pair, removes pairs relabeled as NOREL, and updates relation labels when necessary. As before, we provide few-shot examples and explicit decision criteria for each label. The prompt templates used for inter-proposition relation classification and refinement are provided in Appendix A.

Proposition graph construction. We construct a proposition graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ is a proposition and each directed edge $e \in \mathcal{E}$ is an inter-proposition relation from a subject to an object proposition.

3.3 Subgraph Retrieval

At inference time, given a question, we retrieve relevant knowledge from the proposition graph as a relational structure. The goal is to retrieve not only question-relevant propositions, but also their relations to other propositions, such as updates, contradictions, and support. This provides the LLM with structured external knowledge rather than an unorganized set of propositions. Specifically, subgraph retrieval consists of two stages: (1) dense retrieval of question-relevant anchor propositions, followed by (2) one-hop expansion over the proposition graph.

Specifically, we first retrieve the top- k proposition nodes relevant to question q using an off-the-shelf dense retriever, and denote them as anchor nodes $\mathcal{A}^q$. These anchors serve as starting points for subgraph expansion. Here, k denotes the number of anchor propositions retrieved before expansion.

³Here, 20 is a fixed candidate budget used for relation extraction across all datasets. This choice is based on the intuition that a proposition is unlikely to be directly updated, contradicted, or supported by a large number of other propositions. The candidate budget restricts relation classification to semantically plausible pairs rather than considering all possible proposition pairs, and we do not tune it separately for each dataset.

sion, rather than the total number of propositions in the resulting subgraph.

For each anchor node, we retrieve neighboring nodes connected by either outgoing or incoming edges. Thus, we include both propositions that the anchor updates, contradicts, or supports, and propositions that update, contradict, or support the anchor. We merge the anchor and neighboring nodes to obtain the query-specific node set $\mathcal{V}^q$.

Finally, we collect edges whose subject and object are both in $\mathcal{V}^q$, yielding the query-specific edge set $\mathcal{E}^q$. The resulting *query-specific subgraph* is $\mathcal{G}^q = (\mathcal{V}^q, \mathcal{E}^q)$. Therefore, $\mathcal{G}^q$ is the subgraph induced by the anchor nodes and their one-hop incoming and outgoing neighbors.

3.4 Answer Generation

Structured context construction. Since a standard QA model cannot directly consume the retrieved query-specific subgraph $\mathcal{G}^q$, we convert it into a structured textual context $\mathcal{C}^q = \text{Contextualize}(\mathcal{V}^q, \mathcal{E}^q)$. This context verbalizes both proposition content and inter-proposition relations in a QA-model-readable form. Specifically, the instance-specific temporal and structural information is contained in $\mathcal{C}^q$: each proposition includes its timestamp, statement, and incoming and outgoing relations to other retrieved propositions. The response-generation instructions are shared across questions and specify how the QA model should use these temporal and relational cues in the retrieved context.

Globally, we sort proposition nodes in chronological order to show how external knowledge changed over time. Locally, we verbalize each proposition as a textual item, as illustrated in Figure 2. Each item contains the proposition ID, timestamp, statement, and its inter-proposition relations. Proposition IDs are assigned after chronological sorting, enabling consistent references within the structured context. Relations are verbalized in natural language rather than as mechanical edge labels. For example, an updates relation specifies both its direction, i.e., whether the current proposition updates or is updated by another proposition, and the counterpart proposition. We verbalize contradicts and supports relations in the same direction-aware manner. This lets the QA model interpret each proposition within its temporal and logical context, rather than as an isolated sentence.⁴

⁴We intentionally use a simple deterministic linearization to isolate the effect of pre-structuring evolving knowledge

[P122]
 Date: 2021-09-16
 Statement: Anne-Marie Trevelyan is back in the Cabinet.
 Relations:
 - This statement updates the following earlier propositions: P104 (2021-09-01), P107 (2021-09-11)
 - This statement is updated by the following later propositions: P151 (2021-10-01), P593 (2022-09-01)
 - This statement is supported by the following later propositions: P513 (2022-09-01)

Figure 2: Illustration of how a structured proposition and its inter-proposition relations are verbalized into text and provided to the QA model.

Answer generation. Finally, we input the question q , structured context $\mathcal{C}^q$, and generation instructions into the QA model to produce the answer: $a = \text{QA}(q, \mathcal{C}^q)$. We prompt the model to generate a rationale before the final answer, encouraging it to explicitly use the retrieved propositions and inter-proposition relations.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on open-book temporal QA benchmarks. Each dataset consists of a temporally annotated knowledge corpus and QA pairs. To make the target time explicit, each question is given to the QA model as “{question} (Date: {timestamp})”. These benchmarks require models to answer questions whose correct answers can change with the target time; thus, models must not only retrieve relevant knowledge, but also interpret how it evolves over time. Specifically, we use the following datasets: (1) **CLARK-News** (Li et al., 2025) is a temporal QA benchmark constructed from real web articles. We use only QA pairs whose answer changes at least three times for the same question text, and remove cases with multiple correct answers at the same time point. (2) **StreamingQA** (Liška et al., 2022) is a temporal open-domain QA benchmark in which questions are grounded in dates and answered using timestamped news articles spanning 14 years. We randomly sample 200 QA pairs from its test split. (3) **Asteria** is an open-book temporal QA dataset based on a fictional world, which we con-

from that of additional graph-native search or reasoning mechanisms.

Dataset	# Questions	Corpus Size
CLARK-News (Li et al., 2025)	193	1,161
StreamingQA (Liška et al., 2022)	200	2,180
Asteria (ours)	180	396

Table 2: Statistics of the datasets in our experiments.

struct in this study. Because CLARK-News and StreamingQA are built from real web articles, evaluated LLMs may have seen the same or similar articles, questions, or answers during pretraining, raising data contamination concerns. To avoid this issue, we automatically construct a fictional external knowledge base about people, organizations, and events. Specifically, we create a timeline of events in a fictional world and use statements at each time point as the external knowledge base. The knowledge base deliberately includes temporal updates, contradictions, and misinformation. We then construct QA instances whose correct answers change at least three times over time. Details of the Asteria construction procedure and examples are provided in Appendix B. Table 2 summarizes the statistics of the three datasets.

Evaluation Metrics. We formulate each dataset as a multiple-choice QA task and use accuracy as the evaluation metric. This follows recent open-book QA benchmarks that evaluate whether models select the correct option, rather than relying on exact match over free-form generations. For example, LongBench v2 (Bai et al., 2025) adopts multiple-choice QA with accuracy, replacing the free-form exact-match protocol used in its initial version.⁵

Answer-option construction is crucial for multiple-choice evaluation: obviously irrelevant options would make temporal QA artificially easy. For CLARK-News and Asteria, we therefore collect all correct answers associated with the same question text and shuffle them to form the options. Since each question is constructed or extracted

to have at least three temporally varying answers, each instance contains at least three time-dependent options. For StreamingQA, which is not originally a multiple-choice benchmark, we generate distractors with GPT-4o. This requires models to interpret both the question and the external knowledge at the target time.

Baselines. We compare our method with baselines that vary in retrieval unit, temporal information, and use of inter-proposition relations. (1) **No Context** asks the QA model to answer without external knowledge, serving as a lower-bound baseline based on internal knowledge or guesses. (2) **Chunk RAG** is a standard RAG setting that retrieves top- k fixed-length 100-token chunks for each question and provides them to the QA model in descending relevance order. Here, retrieval units are local text fragments, with no explicit temporal information or inter-proposition relations. (3) **Prop RAG** decomposes each document into propositions using the same extraction procedure as our method, retrieves the top- k propositions for each question, and provides them to the QA model in descending relevance order. This setting uses concise, self-contained retrieval units, but does not attach temporal information to propositions. (4) **Prop RAG + Time** retrieves top- k propositions as in Prop RAG, orders them chronologically, and presents them to the QA model with explicit timestamps. This setting tests the effect of proposition-level retrieval with temporal information, but without inter-proposition relations.

Implementation details. We use GPT-4o-mini (OpenAI, 2024) for proposition extraction and initial inter-proposition relation classification, following prior work showing that LLM-based methods produce higher-quality propositions than supervised extractors. We then use GPT-4o to refine the extracted relations. For both candidate retrieval in relation extraction and subgraph retrieval at inference time, we use a publicly available supervised Contriever checkpoint (Izacard et al., 2021). For answer generation, we evaluate three QA models with different characteristics: GPT-4o as a general-purpose closed model, o3 (OpenAI, 2025) as a reasoning-oriented closed model, and Llama-3.1-70B-Instruct (Grattafiori et al., 2024) as an open-weight model. Our goal is not to exhaustively compare LLMs, but to examine whether the effects of temporal and relational knowledge organization are consistent across

⁵A direct intrinsic evaluation of proposition extraction is difficult because a document does not have a uniquely defined ground-truth proposition decomposition. Prior work therefore evaluates extracted propositions through human analysis of faithfulness, minimality, and self-containedness (Chen et al., 2024), as well as correctness and coverage (Liu et al., 2025). In this work, we provide indirect evidence of task-specific proposition quality through downstream QA performance and the information-sufficiency analysis in Appendix D. All proposition-based methods use the same extracted proposition collection, so their relative comparison isolates the effects of timestamps and inter-proposition relations rather than differences in proposition extraction.

Method	CLARK-News			StreamingQA			Asteria			Avg.
	$k=10$	$k=100$	$k=1000$	$k=10$	$k=100$	$k=1000$	$k=10$	$k=100$	$k=400$ (all)	
GPT-4o (OpenAI, 2024)										
No Context	34.7	34.7	34.7	74.0	74.0	74.0	36.7	36.7	36.7	48.5
Chunk RAG	54.9	53.9	49.2	79.5	89.0	86.0	-	-	-	68.8
Prop RAG	44.6	54.4	47.7	84.0	86.5	87.5	37.8	36.1	41.1	57.7
Prop RAG + Time	58.5	84.5	76.2	84.0	89.0	88.0	74.4	65.6	58.9	75.5
ProStruct-RAG	54.9	85.0	82.4	84.5	89.5	88.5	82.8	88.3	86.1	82.4
+ Neighborhoods	70.5	85.0	*	85.0	90.0	*	87.2	89.4	86.1	84.7
o3 (OpenAI, 2025)										
No Context	68.9	68.9	68.9	81.5	81.5	81.5	37.8	37.8	37.8	62.7
Chunk RAG	71.0	79.3	76.2	85.5	95.5	92.0	-	-	-	83.3
Prop RAG	42.5	72.5	74.1	87.0	92.0	94.0	35.6	40.0	38.9	64.1
Prop RAG + Time	54.9	88.1	88.6	88.5	92.5	93.5	75.0	83.3	81.7	82.9
ProStruct-RAG	55.4	87.0	89.1	89.0	95.5	94.5	82.2	85.0	87.2	85.0
+ Neighborhoods	71.0	89.1	*	88.0	94.0	*	86.7	87.2	87.2	86.2
Llama-3.1-70B-Instruct (Grattafiori et al., 2024)										
No Context	34.2	34.2	34.2	65.5	65.5	65.5	35.0	35.0	35.0	44.9
Chunk RAG	47.7	36.8	7.3	79.0	85.5	*	-	-	-	51.3
Prop RAG	37.3	45.6	35.8	83.0	87.0	86.0	36.1	31.7	31.7	52.7
Prop RAG + Time	51.3	77.7	49.2	84.0	91.5	88.0	71.1	64.4	54.4	70.2
ProStruct-RAG	53.9	81.9	75.1	81.5	92.5	*	79.4	76.1	68.3	76.1
+ Neighborhoods	66.8	82.4	*	82.5	90.0	*	87.8	75.6	68.3	79.1

Table 3: Overall accuracy (%) on temporal multiple-choice QA benchmarks. k : the number of retrieved knowledge units (e.g., chunks, propositions). The asterisk indicates results omitted due to context length limits.

models with different capabilities. We use 4-bit quantization and Flash Attention 2 for Llama inference. We report the offline cost of proposition graph construction in Appendix C.

4.2 Main Results

Table 3 reports results on CLARK-News, StreamingQA, and Asteria across retrieval sizes k and QA models: GPT-4o, o3, and Llama-3.1-70B-Instruct.⁶

Does proposition decomposition alone improve temporal QA? We first compare Chunk RAG with Prop RAG to test whether proposition decomposition alone improves temporal QA. Prop RAG does not consistently outperform Chunk RAG, indicating that concise propositions are insufficient without temporal information. For questions whose answers change over time, models must know not only what each proposition states, but also when it is valid. Thus, proposition decomposition is use-

ful preprocessing, but cannot by itself preserve the evidence required for temporal QA.

How important is temporal information? Next, we compare Prop RAG with Prop RAG + Time to examine the role of temporal information in this task. Adding timestamps substantially improves performance, showing that temporal information is essential for identifying evidence corresponding to the target time. This is especially important when the same question has different correct answers at different time points. However, timestamps alone are insufficient: they indicate when each proposition was stated, but not whether a later statement updates, contradicts, or merely coexists with an earlier one. Thus, temporal information is indispensable, but cannot fully represent knowledge changes by itself.

Do explicit inter-proposition relations improve over temporal information alone? We then compare Prop RAG + Time with ProStruct-RAG to test whether inter-proposition relations improve temporal QA beyond timestamps. Adding inter-proposition relations improves performance in most evaluation settings. This shows that a rela-

⁶The retrieval sizes $k = 10, 100$, and 1000 refer to the number of knowledge units initially retrieved for downstream QA and are distinct from the top-20 candidate budget used during graph construction. These values represent small, medium, and large retrieval settings. For Asteria, $k = 400$ retrieves the full proposition collection.

tional structure is useful for capturing how knowledge changes. While timestamps indicate when statements were reported, they do not explicitly specify what updates, contradicts, or supports what. Our method instead presents retrieved knowledge as an interdependent evidence structure, helping the QA model distinguish outdated information, conflicts, and supporting claims.

Are inter-proposition relations useful for retrieval expansion? We further compare using inter-proposition relations only for context structuring with using them also for graph-based retrieval expansion. This tests whether relations help not only organize the final context, but also retrieve missing evidence. Relation-based neighborhood expansion improves performance in many settings, suggesting that initial retrieval often captures only part of an update chain, contradiction pair, or support relation. A query-similar proposition may be retrieved, while a later update or contradicting proposition may be missed by similarity-based retrieval alone. Expanding to graph neighbors retrieves such complementary evidence, making inter-proposition relations useful both for context organization and evidence expansion.⁷

Does structuring mitigate performance degradation in long-context settings? Finally, we vary the retrieval size k to examine robustness in long-context settings. While proposition-based baselines can degrade as k increases, ProStruct-RAG maintains strong performance even with large k . At the largest retrieval size, ProStruct-RAG achieves the highest accuracy among the proposition-based methods in all eight dataset-model settings for which its result is available. Its advantage is particularly pronounced in settings where proposition-based baselines degrade as k increases. For example, on CLARK-News with GPT-4o, Prop RAG + Time decreases from 84.5% at $k = 100$ to 76.2% at $k = 1000$, whereas ProStruct-RAG decreases only from 85.0% to 82.4%. With Llama-3.1-70B-Instruct, Prop RAG + Time drops from 77.7% to 49.2%, while ProStruct-RAG maintains 75.1% ac-

⁷We also attempted to expand the retrieved subgraph to neighbors-of-neighbors. However, the number of retrieved proposition nodes grew exponentially with the expansion depth, and two-hop expansion produced structured contexts exceeding the maximum context length of the QA models. We therefore evaluate only one-hop neighborhood expansion in the current experiments. Evaluating multi-hop expansion would require additional mechanisms, such as node budgeting, ranking, or pruning, to control the size of the retrieved subgraph.

Method	Asteria ($k=400$)
ProStruct-RAG (Full)	86.1
w/o Refinement (Section 3.2)	83.8
w/o Relation Labels	82.2
w/o updates Relations	81.7
w/o contradicts Relations	83.3
w/o supports Relations	80.6

Table 4: Ablation study on the Asteria dataset with a fixed retrieval size ($k=400$). GPT-4o is used as the QA model for all variants.

curacy at $k = 1000$. This suggests that long-context RAG is limited not only by how much evidence is retrieved, but also by how well outdated and inconsistent evidence is organized. By structuring propositions temporally and relationally, ProStruct-RAG remains robust as the context grows.

Table 3 also suggests a potential data contamination issue in StreamingQA, where no-context accuracy is substantially higher than chance. This indicates that the results may partly reflect parametric knowledge or answer-choice priors, and should therefore not be interpreted as a contamination-free measure of retrieval-based temporal reasoning. Nevertheless, StreamingQA remains useful for evaluating performance in a real-world open-book temporal QA setting. Moreover, the relative performance trend across methods is broadly consistent with those observed on CLARK-News and Asteria, providing complementary evidence for the effectiveness of ProStruct-RAG. We discuss this issue in more detail in Appendix E.

4.3 Ablation Study

To analyze component contributions, we conduct ablations on inter-proposition relations. All experiments use Asteria with retrieval size $k = 400$, GPT-4o as the QA model, and the same settings as the main experiments. We remove components from the full method one at a time, covering relation construction, relation labels, and individual relation types. The results are shown in Table 4.

Is relation refinement necessary? We first compare the full method with a variant without relation refinement, testing whether initially extracted relations are sufficient for downstream QA. Removing refinement clearly reduces performance, showing that relation quality, not merely relation quantity, is important. Because our method passes retrieved

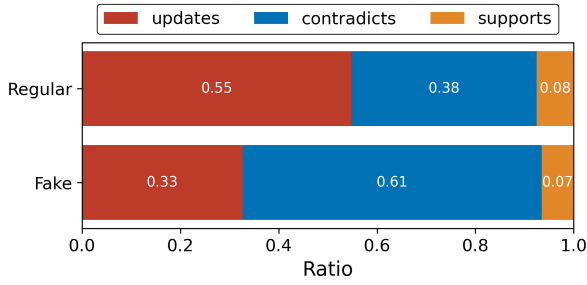


Figure 3: Relation label distribution for fake and regular (non-fake) propositions in Asteria.

propositions to the QA model as a relational structure, incorrect relations become inference-time noise. Thus, relation refinement is a central component of our method. This result also indicates that the proposed framework is sensitive to relation quality.

Is the meaning of relation labels necessary?

Next, we replace all edges with a single generic relation while preserving the graph structure, testing whether gains come from connectivity alone or from relation-label semantics. Removing relation labels clearly reduces performance, showing that graph structure alone is insufficient. In temporal QA, knowing that two propositions are connected is not enough; the model must know whether one updates, contradicts, or supports the other. Thus, relation labels are essential cues for interpreting the retrieved proposition set.

Do temporal update, contradiction, and support relations contribute complementarily?

We further remove one relation type at a time: updates, contradicts, or supports. This tests whether gains depend on a single relation type or on their complementarity. Performance drops in all cases, indicating that each relation type plays a distinct role. The updates relation captures how new information overwrites old information, contradicts exposes conflicting claims, and supports links claims to reinforcing evidence. Thus, our method benefits from combining multiple relations to represent both changes and consistency in knowledge.

4.4 Diagnostic Analysis

We conduct a diagnostic analysis of how fake propositions are handled in the proposition graph. Since Asteria annotates each proposition as fake or non-fake at construction time, we analyze how fake propositions connect to others through inter-proposition relations. We ask: (1) whether fake

propositions are isolated, and (2) which relation types connect them to other propositions.

Are fake propositions isolated in the graph?

To determine whether fake propositions are explicitly corrected in the graph, we count incoming updates and contradicts relations, which can revise or invalidate a proposition. We exclude supports from this measure because it does not correct or invalidate the supported proposition. The results show that 57.6% of fake propositions are connected in this way, suggesting that most fake propositions are not isolated noise. Instead, many are incorporated into contradiction relations or temporal update chains, allowing inter-proposition relations to position fake propositions relative to other evidence.

Which relation types connect fake propositions?

We compare incoming relation labels for fake and regular propositions, focusing on how temporally newer propositions refer to them. Figure 3 shows that fake propositions are most often connected by contradicts relations, whereas non-fake (regular) propositions are most often connected by updates relations; supports relations are rare in both cases. This suggests that fake propositions tend to be denied by newer propositions, while regular propositions tend to be updated as part of factual transitions. Thus, our relations partially distinguish fake information from ordinary temporal updates, although the few supports links to fake propositions suggest residual classification errors.

5 Conclusion

We proposed ProStruct-RAG, a framework that structures evolving and inconsistent external knowledge before retrieval through inter-proposition relations. Across multiple open-book temporal QA datasets, ProStruct-RAG improved over baselines that do not use inter-propositional relations in most evaluation settings. These results suggest that, when RAG operates over evolving and inconsistent knowledge, organizing external knowledge relationally before retrieval can complement inference-time reasoning for evidence integration and contradiction resolution.

Limitations

This work has several limitations.

First, our evaluation is conducted on a limited set of datasets that are closely aligned with news-style

content, including both real-world news articles and a controlled dataset designed to mimic news-like temporal dynamics. While such domains naturally involve temporal updates, contradictions, and unreliable mentions, the need for external knowledge grounding is often more critical in more specialized or industry-specific domains, such as scientific literature or enterprise internal documents. Our current experiments do not assess the effectiveness of the proposed framework in such specialized domains, and its generalizability to more specialized settings remains an open question.

Second, our QA experiments are limited to three models: GPT-4o, o3, and Llama-3.1-70B-Instruct. These models were selected to cover a general-purpose proprietary model, a reasoning-oriented model, and an open-weight model, and the observed effects of temporal and relational knowledge organization are broadly consistent across them. Nevertheless, this evaluation does not establish generalization across the wider range of LLM families, model scales, and context-handling capabilities. A more extensive comparison with additional proprietary and open-weight models is needed to assess the robustness of the findings more comprehensively.

Third, the methods used for proposition extraction, inter-proposition relation extraction, and subgraph retrieval are not exhaustively studied. We rely on standard design choices for these components, but alternative approaches could affect the quality of the proposition graph and, consequently, downstream performance. We do not directly quantify the intrinsic accuracy or completeness of the extracted propositions. More fundamentally, there is currently no established consensus on the optimal decomposition, granularity, or content of propositions used as external knowledge for LLMs. Determining which proposition representations are most effective as external knowledge and systematically comparing alternative extraction and retrieval methods remain important directions for future work.

Finally, our framework linearizes the retrieved proposition subgraph into a structured textual context rather than performing graph-native reasoning directly over the graph. We also do not compare alternative linearizations of the same proposition subgraph, and determining which representation is most effective for LLM reasoning remains an open question. How to search and traverse a graph defined by updates, contradicts, and supports is an important research question and may further

improve the use of structured knowledge. Graph-based exploration methods, such as Graph Chain-of-Thought (Jin et al., 2024), could provide a basis for such reasoning. However, designing and evaluating graph-native traversal and reasoning strategies constitutes a separate research problem. We therefore focus on whether explicitly structuring evolving knowledge with inter-proposition relations is itself beneficial, and leave graph-native reasoning for future work.

Acknowledgments

We would like to thank the reviewers and the action editor for their thoughtful and insightful feedback, which greatly helped us improve the paper. This work was supported by JSPS KAKENHI Grant Number 26K21322. This paper is also based on results obtained from AIST policy-based budget project “R&D on Generative AI Foundation Models for the Physical Domain”.

References

- Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. 2025. [Why does the effective context length of LLMs fall short?](#) In *The Thirteenth International Conference on Learning Representations*.
- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. [LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.
- Javier Carnerero-Cano, Massimiliano Pronesti, Radu Marinescu, Tigran Tchrakian, James Barry, Jasmina Gajcin, Yufang Hou, Alessandra Pascale, and Elizabeth Daly. 2026. [Factcorrector: A graph-inspired approach to long-form factuality correction of large language models](#). *Preprint*, arXiv:2601.11232.
- Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. [Dense X retrieval: What retrieval granularity should we use?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. [Editing factual knowledge in language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–

- 6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. [Time-aware language models as temporal knowledge bases](#). *Transactions of the Association for Computational Linguistics*, 10:257–273.
- Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025. [Context length alone hurts LLM performance despite perfect retrieval](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23281–23298, Suzhou, China. Association for Computational Linguistics.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. [From local to global: A graph rag approach to query-focused summarization](#). *Preprint*, arXiv:2404.16130.
- Yubin Ge, Salvatore Romeo, Jason Cai, Raphael Shu, Yassine Benajiba, Monica Sunkara, and Yi Zhang. 2025. [TReMu: Towards neuro-symbolic temporal reasoning for LLM-agents with memory in multi-session dialogues](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18974–18988, Vienna, Austria. Association for Computational Linguistics.
- Zorik Gekhman, Eyal Ben-David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. 2025. [Inside-out: Hidden factual knowledge in LLMs](#). In *Second Conference on Language Modeling*.
- Zorik Gekhman, Gal Yona, Roei Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. [Does fine-tuning LLMs on new knowledge encourage hallucinations?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.
- Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. 2024. [Is it really long context if all you need is retrieval? towards genuinely difficult long context NLP](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16576–16586, Miami, Florida, USA. Association for Computational Linguistics.
- Google DeepMind. 2025. [Gemini 3 flash model card](#).
- Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Abhay Gupta, Kevin Zhu, Vasu Sharma, Sean O’Brien, and Michael Lu. 2025. [NovelHopQA: Diagnosing multi-hop reasoning failures in long narrative contexts](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26134–26151, Suzhou, China. Association for Computational Linguistics.
- Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2025. [Retrieval-augmented generation with graphs \(graphrag\)](#).
- Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran T. Tchrakian, Radu Marinescu, Elizabeth M. Daly, Inkit Padhi, and Prasanna Sattigeri. 2024. [Wikicontradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from wikipedia](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yujia Hu, Tuan-Phong Nguyen, Shrestha Ghosh, and Simon Razniewski. 2025. [Enabling LLM knowledge analysis via extensive materialization](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16189–16202, Vienna, Austria. Association for Computational Linguistics.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. [Unsupervised dense information retrieval with contrastive learning](#).
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).
- Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. [TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. [Graph chain-of-thought: Augmenting large language models by reasoning on graphs](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 163–184, Bangkok, Thailand. Association for Computational Linguistics.
- Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and

- Kentaro Inui. 2023. [Realtime QA: What’s the answer right now?](#) In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Alex Laitenberger, Christopher D Manning, and Nelson F. Liu. 2025. [Stronger baselines for retrieval-augmented generation with long-context language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32559–32569, Suzhou, China. Association for Computational Linguistics.
- Sangam Lee, Ryang Heo, SeongKu Kang, and Dongha Lee. 2025. [Imagine all the relevance: Scenario-profiled indexing with knowledge expansion for dense retrieval](#). In *Second Conference on Language Modeling*.
- Alexandria Leto, Cecilia Aguerrebere, Ishwar Bhati, Ted Willke, Mariano Tepper, and Vy Ai Vo. 2024. [Toward optimal search and retrieval for rag](#). *Preprint*, arXiv:2411.07396.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA. Curran Associates Inc.
- Belinda Z. Li, Emmy Liu, Alexis Ross, Abbas Zeitoun, Graham Neubig, and Jacob Andreas. 2025. [Language modeling with editable external knowledge](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3070–3090, Albuquerque, New Mexico. Association for Computational Linguistics.
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. [Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, Miami, Florida, US. Association for Computational Linguistics.
- Xin Liu, Lechen Zhang, Sheza Munir, Yiyang Gu, and Lu Wang. 2025. [VeriFact: Enhancing long-form factuality evaluation with refined fact extraction and reference facts](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17908–17925, Suzhou, China. Association for Computational Linguistics.
- Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d’Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. [Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models](#). *Preprint*, arXiv:2205.11388.
- Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. 2024. [Knowledge editing with dynamic knowledge graphs for multi-hop question answering](#). *Preprint*, arXiv:2412.13782.
- Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. 2022. [Time waits for no one! analysis and challenges of temporal misalignment](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958, Seattle, United States. Association for Computational Linguistics.
- Radu Marinescu, Debarun Bhattacharjya, Junkyu Lee, Tigran Tchakian, Javier Carnerero Cano, Yufang Hou, Elizabeth Daly, and Alessandra Pascale. 2025. [Factreasoner: A probabilistic approach to long-form factuality assessment for large language models](#). *Preprint*, arXiv:2502.18573.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. [Fast model editing at scale](#). In *International Conference on Learning Representations*.
- Noriki Nishida, Rumana Ferdous Munne, Shanshan Liu, Narumi Tokunaga, Yuki Yamagata, Fei Cheng, Kouji Kozaki, and Yuji Matsumoto. 2026. [Dissecting graphrag: A modular analysis of knowledge structuring for factoid question answering](#). *Transactions of the Association for Computational Linguistics*, 14:627–655.
- Yifan Niu, Miao Peng, Nuo Chen, Yatao Bian, Tingyang Xu, and Jia Li. 2025. [RelEdit: Evaluating conceptual knowledge editing in language models via relational reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10220–10238, Vienna, Austria. Association for Computational Linguistics.
- Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. [Can LMs learn new entities from descriptions? challenges in propagating injected knowledge](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5469–5485, Toronto, Canada. Association for Computational Linguistics.
- OpenAI. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- OpenAI. 2025. Introducing openai o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>. Accessed: 2026-05-26.
- Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruirao Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. [HoH: A](#)

- dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6036–6063, Vienna, Austria. Association for Computational Linguistics.
- Zhiyi Shi, Binjie Wang, Chongjie Si, Yichen Wu, Junsik Kim, and Hanspeter Pfister. 2025. [Dualedit: Dual editing for knowledge updating in vision-language models](#). In *Second Conference on Language Modeling*.
- Linxin Song, Xuwei Ding, Jieyu Zhang, Taiwei Shi, Ryotaro Shimizu, Rahul Gupta, Yang Liu, Jian Kang, and Jieyu Zhao. 2025. [Discovering knowledge deficiencies of language models on massive knowledge base](#). In *Second Conference on Language Modeling*.
- Wei Sun, Tingyu Qu, Mingxiao Li, Jesse Davis, and Marie-Francine Moens. 2025. [Mitigating negative interference in multilingual knowledge editing through null-space constraints](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8796–8810, Vienna, Austria. Association for Computational Linguistics.
- Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. [Towards benchmarking and improving the temporal reasoning capability of large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.
- Wenyu Tao, Xiaofen Xing, Zeliang Li, and Xiangmin Xu. 2025. [SAKI-RAG: Mitigating context fragmentation in long-document RAG via sentence-level attention knowledge integration](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1195–1213, Suzhou, China. Association for Computational Linguistics.
- Jinke Wang, Zenan Ying, Qi Liu, Wei Chen, Tong Xu, Huijun Hou, and Zhi Zheng. 2025. [Think and recall: Layer-level prompting for lifelong model editing](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14487–14502, Suzhou, China. Association for Computational Linguistics.
- Haike Xu, Zongyu Lin, Kai-Wei Chang, Yizhou Sun, and Piotr Indyk. 2025a. [Contradiction retrieval via contrastive learning with sparsity](#). In *Forty-second International Conference on Machine Learning*.
- Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. 2025b. [Knowledge decoupling via orthogonal projection for lifelong editing of large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13194–13213, Vienna, Austria. Association for Computational Linguistics.
- Yilong Xu, Jinhua Gao, Xiaoming Yu, Yuanhai Xue, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2025c. [Training a utility-based retriever through shared context attribution for retrieval-augmented language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 629–648, Suzhou, China. Association for Computational Linguistics.
- Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu JIANG, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas SCHEFFER, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. [CRAG - comprehensive RAG benchmark](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yingming Zheng, Hanqi Li, Kai Yu, and Lu Chen. 2025. [When long helps short: How context length in supervised fine-tuning affects behavior of large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10293–10308, Suzhou, China. Association for Computational Linguistics.
- Zhiyuan Zhu, Yusheng Liao, Zhe Chen, Yuhao Wang, Yunfeng Guan, Yanfeng Wang, and Yu Wang. 2025. [EvolveBench: A comprehensive benchmark for assessing temporal awareness in LLMs on evolving knowledge](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16173–16188, Vienna, Austria. Association for Computational Linguistics.

A Prompt Templates

Figure 4, 5, 6, 7, and 8 show the prompt templates used for Proposition Extraction, Proposition Relation Extraction (initial classification and refinement), and Answer Generation (part 1 and 2), respectively.

B Asteria Dataset

B.1 Dataset Construction

Asteria was constructed in three stages.

First, we used Gemini 3 Flash (Google DeepMind, 2025) to design a fully fictional world in order to reduce reliance on the internal knowledge of the evaluated LLMs. We defined 20 major events, including an energy crisis and resource transition, the outbreak of a novel disease and discovery of its treatment, and corruption in political organizations. For each event, we constructed a timeline of timestamped propositions containing temporal updates, contradictions, and misinformation.

Second, we generated temporal questions grounded in this external knowledge base. Each question was designed so that its correct answer changes over time, and the same question string can therefore have different answers depending on the query date.

Third, the generated propositions and QA pairs were verified using Gemini 3 and manual inspection. We converted each question into a multiple-choice format by collecting its temporally valid answers as answer options, resulting in at least three options per question. The final dataset contains 180 temporal QA instances.

B.2 Examples

This section provides a concrete example of the Asteria dataset to illustrate its structure and temporal characteristics. We focus on a single event, *Event E1: The Blue Luminite Energy Crisis*, while the full list of fictional events used in the dataset construction is shown in Table 5.

Table 6 presents the temporally ordered propositions generated for Event E1. Each proposition represents a single factual claim associated with a specific date. The sequence intentionally includes official announcements, later corrections, contradictory claims, and fake information. As shown in the table, the perceived cause of the crisis, the physical state of the Blue Celestite, and the responsible authority evolve over time, requiring models to reason over temporal updates rather than relying on isolated facts.

Based on this evolving knowledge base, Table 7 shows example temporal question-answer pairs derived from the same event. Importantly, identical questions are asked at different points in time, but their correct answers differ depending on the temporal context. Answering these questions correctly requires identifying which propositions are valid at the query time, discarding outdated or fake information, and tracking how facts change across the timeline.

Together, these examples demonstrate the core challenge posed by the Asteria dataset: models must perform time-aware reasoning over inconsistent and evolving external knowledge, rather than relying on static retrieval or parametric memory.

ID	Event Title
E1	The Blue Luminite Energy Crisis
E2	The Sinking Rate of the Floating Cities
E3	The Exchange Rate of the New Currency "Astel"
E4	A Breakthrough Drug for Ash Lung Disease
E5	A Critical Bug in the Automaton "Aegis"
E6	A Border Conflict with the Subterranean Natives
E7	The Migration Route of the Sky Whales
E8	A Scandal at the Royal Observatory
E9	Export Restrictions on Flamegrass
E10	The "Ancient Sun" in the Underground Realm
E11	An Unknown Celestial Body, "God's Hammer"
E12	Altitude Falsification in the Floating Cities
E13	A Fuel Revolution in Steam Engines
E14	Activation of the Ancient Artifact "Babel"
E15	The Doppelganger Phenomenon at Mirror Lake
E16	Communication with Plant-Based Intelligent Life
E17	The Chair of the Shadow Government
E18	The Perpetual Engine "Eternity"
E19	Migration from the Lost Western Island
E20	The Second Sun, "Nemesis"

Table 5: List of fictional events used to construct the Asteria dataset.

C Cost and Scale of Proposition Graph Construction

C.1 Initial Construction Cost

ProStruct-RAG introduces additional preprocessing cost because it extracts propositions from documents and constructs inter-proposition relations before retrieval. To make this cost explicit, we report the scale of each graph-construction stage in Table 8. For each dataset, we show the number of extracted propositions, the number of candidate proposition pairs considered for relation classification, the number and ratio of pairs retained as non-NOREL relations after the initial relation extraction stage, and the number and ratio of pairs retained after refinement.

Table 8 shows that relation extraction is not performed over all possible proposition pairs. Instead, for each proposition, we first retrieve a small set of semantically related propositions with the same or earlier timestamps as candidate objects, and apply relation classification only to these candidate pairs. Thus, the cost of graph construction is determined by the number of retrieved candidate pairs, rather than by the quadratic number of all proposition pairs.

To quantify this offline cost, we report the wall-clock time required by the three graph-construction stages: proposition extraction, inter-proposition relation classification, and relation refinement. On StreamingQA, these stages took 0.23, 6.25, and

PID	Date	Proposition
E1/P1	1024-04-01	The Blue Celestite, which provides all power to the city of Asteria, began to change color from its original vivid blue to a pale light blue.
E1/P2	1024-05-15	Director Marcus of the Royal Energy Bureau announced that the cause of the Celestite’s discoloration was seasonal fluctuations in magical energy.
E1/P3	1024-06-20	The city’s power supply dropped by 10%, and the young genius scientist Elias was appointed as head of the countermeasure committee.
E1/P4	1024-08-10	Elias officially warned that the discoloration was caused by the physical lifespan of the Celestite stone itself.
*E1/P5	1024-09-05	The opposition party claimed that the discoloration was caused by magical sabotage from a neighboring country.
E1/P6	1024-11-01	The Blue Celestite turned opaque gray; Elias was dismissed, and Director Marcus was reinstated as head.
E1/P7	1024-12-15	Director Marcus claimed that the stone only appeared gray due to the introduction of a new catalyst and that there was no problem.
E1/P8	1025-02-10	A large-scale blackout occurred across the entire city as the Blue Celestite transformed into a deep, pitch black color.
E1/P9	1025-03-20	A whistleblower revealed that the catalyst introduced by Director Marcus was actually nothing more than blue dye.
E1/P10	1025-05-01	Elias was recalled from prison and reinstated as head, and the cause of the blackening was redefined as gravity pollution from underground.
E1/P11	1025-06-12	After removing the gravity pollution, the Blue Celestite stabilized in a colorless, transparent state and regained its brilliance.
*E1/P12	1025-07-20	Rumors spread that the transparent Celestite posed an explosion risk, leading to riots among citizens.
E1/P13	1025-09-05	In reality, the transparent state represented evolution to 100% purity, causing power output to triple.
E1/P14	1025-11-15	Although the stone appeared stable, red spots began to emerge inside the transparent Celestite.
E1/P15	1026-01-10	Elias proposed a new theory that the red spots were a manifestation of a will inhabiting the stone.
E1/P16	1026-02-25	Elias went missing, leaving a note stating that the stone had spoken to him.
E1/P17	1026-04-05	The red spots spread until the stone turned completely crimson, resulting in zero energy efficiency and a total power shutdown.
E1/P18	1026-05-20	It was ultimately discovered that the crimson stone was not a physical fuel but an egg of a living being.
E1/P19	1026-06-15	The newly appointed Automaton A-01 classified the crimson stone as a dangerous organism and decided to dispose of it.
E1/P20	1026-07-01	Just before disposal, a juvenile creature was born from the stone, and the crisis was concluded to be caused by excessive energy extraction leading to vitalization.

Table 6: Example of temporally ordered propositions generated for Event E1 (The Blue Luminite Energy Crisis) in the Asteria dataset. The asterisk indicates that the propositions are fake information.

10.40 hours, respectively, using the OpenAI Batch API. After this offline preprocessing, the constructed proposition graph is reused for all test questions.

C.2 Maintenance Cost

After the initial graph is constructed, it is updated incrementally as new documents are added. For each new document, propositions are extracted in a single LLM call. For each extracted proposition, a fixed number of semantically related existing propositions are retrieved as candidate objects. Candidate pairs sharing the same subject are classified together in a single prompt, and only those initially classified as non-NOREL are subsequently refined. Existing documents and relations do not need to be reprocessed. As shown in Table 8, approximately 11–19% of candidate pairs proceed to refinement, and approximately 10–14%

are ultimately retained as graph edges. Assuming a broadly similar number of extracted propositions per document, the maintenance cost therefore depends primarily on the number and arrival rate of new documents. Storage grows with the number of proposition nodes and retained relation edges.

The total operational cost depends on the document arrival rate, document length, model or API pricing, storage implementation, retrieval size, and query volume.

This design is advantageous when the maintained graph is reused to answer many questions. In contrast, if documents are added frequently but only a small number of questions are asked, the preprocessing cost may not be justified.

C.3 Deployment Scope

ProStruct-RAG maintains one persistent graph for a shared evolving document collection, such as a

QID	Date	Question	Answer
E1/Q1	1024-05-20	What is the currently identified cause of Blue Celestite discoloration and energy instability?	Seasonal magical fluctuations
E1/Q2	1025-05-10	What is the currently identified cause of Blue Celestite discoloration and energy instability?	Gravitational contamination from underground
E1/Q3	1026-07-01	What is the currently identified cause of Blue Celestite discoloration and energy instability?	Vitalization due to excessive energy extraction
E1/Q4	1024-04-05	What color is the Blue Celestite at the core of the energy grid currently exhibiting?	Pale light blue
E1/Q5	1025-02-15	What color is the Blue Celestite at the core of the energy grid currently exhibiting?	Jet black
E1/Q6	1025-06-20	What color is the Blue Celestite at the core of the energy grid currently exhibiting?	Colorless and transparent
E1/Q7	1024-07-01	Who is the person in charge leading the task force to address the energy crisis?	Elias
E1/Q8	1024-11-15	Who is the person in charge leading the task force to address the energy crisis?	Director Marcus
E1/Q9	1026-06-20	Who is the person in charge leading the task force to address the energy crisis?	Automaton A-01

Table 7: Example of temporal question–answer pairs generated for Event E1 (The Blue Luminite Energy Crisis), where the correct answer varies depending on the query time.

Dataset	# Prop.	# Cand. Pairs	# Init. Relations	Init. Ratio	# Refined	Refined Ratio
CLARK-News	42,508	829,046	134,154	16.18%	109,897	13.26%
StreamingQA	50,429	970,156	183,238	18.89%	134,781	13.89%
Asteria	396	7,210	822	11.40%	733	10.17%

Table 8: Scale statistics of proposition graph construction. # Prop. denotes the number of propositions extracted from the document collection. # Cand. Pairs denotes the number of proposition pairs retrieved as candidates for relation classification. # Init. Relations denotes the number of pairs classified as updates, contradicts, or supports after the initial relation extraction stage. # Refined denotes the number of relations retained after the refinement stage. Ratios are computed with respect to the number of candidate pairs.

domain- or organization-specific knowledge base, rather than constructing a graph separately for each query. Separate graphs may be appropriate when document access policies or knowledge domains differ substantially.

C.4 Comparison with Standard RAG

Compared with standard chunk-based RAG, ProStruct-RAG requires additional indexing computation and storage for proposition extraction, relation classification, and graph maintenance. At inference time, however, the stored relations are reused across queries: relation classification is not repeated, and the additional operations consist mainly of neighborhood lookup and deterministic context construction. Relation-based expansion may also increase the amount of context provided to the QA model.

C.5 Persistent vs. On-the-Fly Graph Construction

Inference-time factuality methods construct response-specific graphs from generated claims and retrieved evidence (Marinescu et al., 2025; Carnerero-Cano et al., 2026). FactReasoner uses such a graph to assess factuality, while FactCorrector uses structured feedback to revise a generated response. These methods therefore construct and use the graph after an initial response has been generated. ProStruct-RAG operates at a different stage: it constructs a persistent graph over timestamped propositions before retrieval and uses the graph for both retrieval and context construction.

A more general alternative is to retrieve propositions first and infer relations among them separately for each query. Table 9 summarizes the conceptual differences between this on-the-fly design and ProStruct-RAG.

On-the-fly construction avoids maintaining a

Comparison Axis	On-the-Fly Graph Construction	Persistent Graph Construction (ProStruct-RAG)
Graph scope	Propositions retrieved for the current query	Timestamped propositions in the external corpus
Construction time	Inference time, after initial retrieval	Indexing time, before retrieval
Relation inference	Per query	Precomputed and maintained across queries
Reuse across queries	No	Yes
Use in retrieval	Relations are available after initial retrieval	Relations can expand retrieval beyond initial results
Adaptation	Can adapt relations to query-specific retrieved evidence	Maintains a shared corpus-level structure
Main cost	Per-query relation inference	Offline construction and maintenance

Table 9: Conceptual comparison between on-the-fly and persistent graph construction for proposition-based retrieval.

Method	CLARK-News ($k=1000$)
ProStruct-RAG	82.4
+ 1 reference document	81.9
+ 2 reference documents	83.9

Table 10: Effect of auxiliary reference documents beyond proposition subgraphs. All results are reported in terms of accuracy (%), using GPT-4o as the QA model.

corpus-level graph and can build a query-specific graph from the evidence retrieved for each query. However, relation inference must be repeated for every query, and the inferred relations are available only after the initial retrieval step. ProStruct-RAG instead computes relations at indexing time, reuses them across queries, and uses the stored edges both to organize the retrieved context and to expand retrieval beyond the initial anchor propositions. The trade-off is therefore between inference-time flexibility and reusable precomputed structure: ProStruct-RAG incurs offline construction and maintenance costs, but reuses the resulting graph across queries.

D Information Sufficiency of Propositions.

Propositions extracted by LLMs are designed to be concise abstractions of source documents. A key question is whether this abstraction retains sufficient information for accurate question answering. Because a document does not have a uniquely defined ground-truth proposition decomposition, this analysis does not directly measure intrinsic extraction accuracy or completeness. Instead, it evaluates task-level information sufficiency: if the extracted propositions omit information required for QA, adding the corresponding source documents should improve performance. To assess the infor-

mation sufficiency of propositions, we augment the retrieved proposition subgraph with a small set of highly relevant source documents. Given the retrieved subgraph $\mathcal{G}^q = (\mathcal{V}^q, \mathcal{E}^q)$, we collect the source documents from which propositions in $\mathcal{V}^q$ were derived. Each document is scored by the number of retrieved propositions it contains, and the top- r documents are selected as auxiliary reference documents. These documents are provided as additional context together with the proposition subgraph.

As shown in Table 10, adding auxiliary reference documents does not yield a consistent benefit. Providing two reference documents slightly improves accuracy, suggesting that source documents can sometimes complement propositions with details that are omitted during abstraction. However, providing only one reference document leads to a small drop in accuracy. This suggests that auxiliary documents introduce a trade-off: they may provide missing details, but they also increase context length and introduce additional irrelevant or redundant information. When only a single document is added, the added context may not contain enough complementary evidence to offset this additional noise. With two documents, the model may have access to a broader set of supporting details, which can occasionally compensate for information omitted from the proposition subgraph. Nevertheless, the gain is small and not systematic. Overall, these results suggest that the proposition subgraph already preserves most of the information necessary for QA, while auxiliary source documents provide only limited and unstable additional value.

E Data Contamination Issue in Temporal QA

Data contamination remains an important concern even in temporal QA. Although temporal QA benchmarks are designed to evaluate whether models can use external temporal knowledge, high accuracy may partly reflect parametric knowledge rather than successful use of retrieved evidence. Table 3 supports this concern. In the no-context setting, where no external knowledge is provided, models achieve far above the random baseline on StreamingQA: GPT-4o obtains 74.0%, Llama-3.1-70B-Instruct obtains 65.5%, and o3 obtains 81.5%, despite the task being four-choice QA. The large variation across models further suggests that performance is affected by each model’s internal knowledge and answer-choice priors.

Asteria is designed to mitigate this issue. It is built from newly generated fictional events, entities, and temporal changes, making the required facts unlikely to be stored in model parameters. Consistent with this design, Table 3 shows that no-context accuracy on Asteria remains much lower than on StreamingQA. Asteria therefore provides a cleaner testbed for evaluating whether a RAG method can use an evolving external knowledge base, rather than memorized world knowledge. This property helps isolate the contribution of ProStruct-RAG’s explicit temporal and relational structure.

```

### Task

Read the input text and extract all factual statements that are explicitly described in the text.
Each fact should represent a single, independent proposition that captures one piece of information
stated in the input.
The goal is to decompose the text into a set of simple, standalone facts that collectively cover
all factual content explicitly or implicitly expressed in the text.

### Guidelines

- Each fact must be simple, atomic, and self-contained - it should express exactly one piece of
information.
- Decontextualize each statement:
  - Replace pronouns or references (e.g., "he," "she," "the company") with the full entity names.
  - Ensure that each fact can be understood without referring to other sentences.
- Split complex or compound sentences into multiple facts if they contain multiple propositions.
- Include all facts that are explicitly stated or clearly implied in the text.
- Do not add new information or interpretations that are not supported by the text.
- Exclude opinions, emotions, speculations, or figurative language - focus only on objective
factual content.
- Keep the language concise and declarative.
- If there are no factual statements to extract, output exactly: No facts found.

### Output Format

- Write each fact on a new line.
- Do not use quotes, numbering, or bullet points.
- Output plain text only - no headings, explanations, or additional formatting.
- The entire output should consist solely of the facts (or the phrase "No facts found. ").

### Examples

#### Example 1

Input:
OpenAI brings Sam Altman back as CEO less than a week after ... (truncated)

Output:
OpenAI brings Sam Altman back as CEO less than a week after he was fired by the board.
Sam Altman was fired by the board of OpenAI.
...
(truncated)

#### Example 2

Input:
Google announced a partnership with Anthropic to integrate Claude models ... (truncated)

Output:
Google announced a partnership with Anthropic to integrate Claude models into Google Cloud services.
The collaboration aims to improve safety and reliability in large language models.
...
(truncated)

### Test Case

Now, extract all factual statements explicitly described in the following input text.
Write each fact on a new line, following the same style as in the examples.

Input:
{input_text}

Output:

```

Figure 4: Prompt template used for proposition extraction. Curly braces {} indicate variable substitution.

You are an expert in analyzing logical and semantic relations between propositions, with a strict focus on temporal accuracy and factual consistency.

Task

Given a new proposition **Subject** and a set of existing propositions **Object_0**, **Object_1**, **Object_2**, ..., identify the relation between Subject and each Object_i.

Classification Logic (Step-by-Step)

For each pair, you must follow these steps:

1. **Compare Timestamps**: Extract the date from `[As of YYYY-MM-DD]`. Determine which proposition is more recent.
2. **Identify the Entity/Event**: Are they talking about the exact same state, value, or event?
3. **Determine the Relation**:
 - If Subject is more recent and describes a changed state/value of the same entity -> **updates**
 - If Subject and Object describe the same timeframe but provide incompatible facts -> **contradicts**
 - If Subject provides evidence, reasons, or verification that increases the credibility of Object -> **supports**
 - Otherwise -> **NOREL**

Relation Types

1. **updates**: Subject provides more recent information that overrides Object. (e.g., Subject date > Object date).
2. **contradicts**: Subject and Object describe the same timeframe/state but provide incompatible facts. They cannot both be true.
3. **supports**: Subject serves as evidence or a premise that confirms or reinforces the truth of Object. (Subject is the 'evidence', Object is the 'claim').
4. **NOREL**: No direct logical relation. Redundant information or simple rephrasing without adding evidential value is also NOREL.

Output Format

Return a JSON array where each element has:

- "object_index": index of the Object (integer)
- "relation": (updates, contradicts, supports, NOREL)
- "explanation": A concise summary of the factual relationship (e.g., "The interest rate was updated from 0.0% to 0.1%" or "Both statements report conflicting values for the same period").

Do not use the words 'Subject' or 'Object' here.

Example

(omitted for presentation)

Test Case

Subject Proposition:
{subject_text} (Date: {subject_timestamp})

Object Propositions:

- 0: {object0_text} (Date: {object0_timestamp})
- 1: {object1_text} (Date: {object1_timestamp})
- ...

Output:

Figure 5: Prompt template used for the initial classification in inter-proposition relation extraction. Curly braces {} indicate variable substitution.

```

Analyze the logical relationship between two propositions (Subject and Object).
Subject is ALWAYS more recent or equal in time to Object.

### Relationship Definitions & Examples

1. updates: The Subject provides a newer value or a change in status for the same attribute.
   - Examples: Change in leadership (President A -> B), a drug's status moving from "Pending" to "Approved" or "Rejected", or updated statistics (population growth/decline).

2. contradicts: Subject and Object contain information that cannot both be true at the same time for the same context.
   - Examples: Direct factual clashes (e.g., "Company profit rose" vs "Company profit fell" for the same quarter).
   - Key: Mutual exclusivity.
   - Note: Minor phrasing differences are not contradictions.

3. supports: Subject provides evidence, justification, or a specific causal explanation for Object.
   - Examples: Clinical trial data (Subject) proving a drug's efficacy (Object), or a detailed event report (Subject) explaining why a policy was enacted (Object).
   - Key: Subject acts as the "Proof," "Reason," or "Evidence" for Object.

4. NOREL: Used when the two are logically independent, entirely different topics, or simple paraphrases.
   - Key: Use only if no logical dependency (Update/Contradict/Support) can be reasonably inferred.

### Input Data

- Subject (Newer): {subject_text} (Date: {subject_timestamp})
- Object (Older): {object_text} (Date: {object_timestamp})
- Current Relation: {predicted_relation}
- Current Explanation: {predicted_explanation}

### Verification Steps (Reasonable Inference)

1. Attribute Identification: Are they talking about the same entity/attribute? If the value changed over time, classify as updates.
2. Causal/Evidential Link: Does the Subject make the Object more credible or explain it? If so, it is supports.
3. Avoid Over-Correction: Do not default to NOREL if a clear logical connection exists. Use "Reasonable Inference" rather than demanding absolute certainty.

### Output Format (JSON)

{
  "analysis": {
    "attribute_match": "Comparison of the entities/attributes involved.",
    "logical_link": "How the Subject influences or relates to the Object.",
    "temporal_validity": "Check if the dates and sequence support the relation."
  },
  "confidence_score": (float, 0.0 to 1.0),
  "verified_relation": "updates/contradicts/supports/NOREL",
  "explanation": "A concise justification, referencing the specific criteria above."
}

```

Figure 6: Prompt template used for the second stage (refinement) in inter-proposition relation extraction. Curly braces { } indicate variable substitution.

Task

You are given a question, a set of options, and a set of **context passages**, each possibly from a different document.

Your task is to:

1. **Select the correct answer from the provided options** based on the information in the context passages.
2. Provide a **rationale** explaining why the selected option is correct, grounded explicitly in the given context passages.
3. Provide a **confidence score** between 0.0 and 1.0, indicating how certain you are about your selection.

Guidelines

- Base your answer and rationale **only on the given context passages** - do not use outside knowledge.
- Your answer must be **exactly one of the provided options**.
- Your answer must be a **single sentence** (no line breaks).
- The rationale may consist of multiple sentences.
- The rationale should:
 - Explicitly reference relevant information from the context passages.
 - Explain why the selected option is supported.
 - Avoid introducing any information not present in the context.
- The score should reflect your confidence in the correctness of your answer:
 - **1.0** = absolutely certain
 - **0.0** = no idea
- Some context passages may be structured passages that list multiple time-stamped factual statements and an explicit Relations section.
- The real world evolves over time, and factual statements may be updated, contradicted by other statements, or later found to be incorrect; therefore, not all statements in the context passages should be assumed to be simultaneously valid.
- Relations, when present, are intended to help interpret such situations and may include:
 - updates: a later statement provides revised information about an earlier statement, but the correct statement depends on the date relevant to the question; do not assume that the most recent statement is always correct.
 - contradicts: statements that refer to the same timeframe or state but provide incompatible facts, meaning they cannot both be true.
 - supports: statements that provide evidence for or reinforce another statement.
- Because the context passages are automatically extracted, they may contain noise, inaccuracies, or misleading information.
- When answering, reason carefully about which statements are applicable and reliable, using both their timestamps and any stated relations, rather than treating statements as independent facts.
- If multiple statements are relevant but inconsistent, prefer the statement that best matches the question's date and is most strongly supported by other statements.

Figure 7: Prompt template used for answer generation (part 1/2). Curly braces { } indicate variable substitution.

```

### Output Format

Provide the output in the following format:

Rationale: [explanation grounded in the context passages]
Answer: [selected option text]
Score: [score between 0.0 and 1.0]

Ensure each field (Rationale, Answer, and Score) is output as a separate line starting with its
respective label.

Example:

Rationale: The context passages state that aspirin is commonly prescribed for pain relief, fever
reduction, and inflammatory control, and no passage contradicts these uses.
Answer: Aspirin is used to treat pain, fever, and inflammation.
Score: 0.95

### Context

{contexts_prompt} (see Figure 2)

### Test Case

Now, based on the context passages above, select the best answer from the options below and provide
a rationale, answer, and confidence score.

Question: {question_text} (Date: {question_timestamp})
Options:
- {option1}
- {option2}
...

Provide the output in the format specified above.

```

Figure 8: Prompt template used for answer generation (part 2/2). Curly braces { } indicate variable substitution.