IntervalRAG: Hierarchical Interval Graphs for Time-Grounded Retrieval-Augmented Generation

Anonymous authors
Paper under double-blind review

Abstract

IntervalRAG is a retrieval-augmented generation framework for questions whose answers depend on when a fact remains valid. Such questions often concern a persistent state, while the supporting evidence is distributed across appointments, departures, and other transitions in a narrative. Retrieving an isolated observation can therefore leave the generator to reconstruct which state applies to the queried period and how its boundaries are established. IntervalRAG organizes this evidence in a two-level temporal graph: macro units identify broad states or phases, and micro units preserve individual facts and transitions together with the temporal information needed to place them in context. Hierarchical edges retain the association between each micro-event and its source context, while local temporal edges expose neighboring evidence for assembling a short chronology. The query first locates a relevant macro context and then guides the selection of nearby micro-events and source passages around the requested period. The resulting context presents the relevant state together with the transitions that explain its temporal boundaries. We evaluate IntervalRAG on TimeQA, TempReason, and ComplexTR, where it obtains accuracies of 60.93%, 91.79%, and 68.39%, respectively. Code and data will be released upon publication.

1 Introduction

Retrieval-augmented generation (RAG) has become a widely adopted paradigm for grounding large language models in external knowledge by retrieving evidence from a corpus and injecting it into the generation context (Lewis et al., 2020; Gao et al., 2024); this retrieval-first design mitigates the static knowledge and hallucination issues of purely parametric models and is effective across open-domain question answering, domain adaptation, and knowledge-intensive reasoning (Siriwardhana et al., 2023; Huang et al., 2025). However, most RAG pipelines remain temporally agnostic: retrieval is driven largely by semantic similarity, with temporal intent treated as an implicit textual feature rather than an explicit constraint enforced during retrieval (Zhu et al., 2025; Han et al., 2025; Gade et al., 2025). This becomes pronounced for time-scoped questions, whose answer depends not only on topical relevance but also on whether the supporting evidence holds within the queried period (Chen et al., 2021; Piryani et al., 2025): when semantically similar evidence spans different eras, a retriever may surface documents that are topically relevant yet temporally incompatible with the query (Abdallah et al., 2025; Reichman & Heck, 2024; Fayyaz et al., 2025; Wang et al., 2025).

Temporal RAG methods incorporate time through event indices, temporal scores, or chronological context ordering (Zhu et al., 2025; Abdallah et al., 2025; Han et al., 2025; Sun et al., 2025; Kim et al., 2025; Gade et al., 2025). A timestamp can locate an observation, whereas an interval also represents the duration of a state and its relation to neighboring transitions (Pustejovsky et al., 2003) (Figure 1).

The key retrieval difficulty arises when the query falls inside a long-running state or between two events that mark a change. Questions about a person's affiliation in a given year illustrate this gap: a semantically similar appointment is not enough, because the relevant state may have started earlier, ended later, or continued through the queried date. Such evidence is often distributed across a narrative as a state together with the events that establish its boundaries, so RAG must account for both what a fact is and the period over which it holds. If retrieval returns only isolated or cross-period

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.

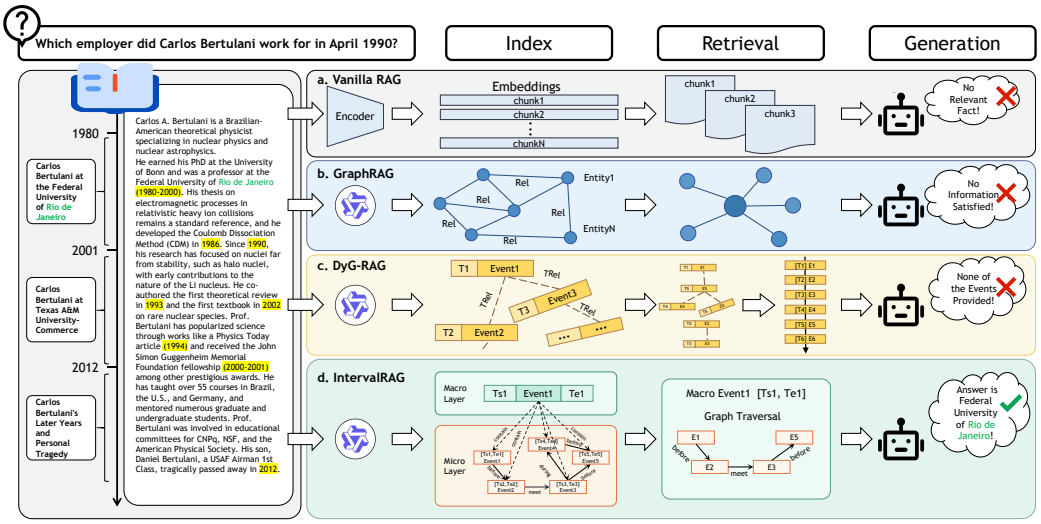


Figure 1: Evidence representations for a time-scoped question. Chunk retrieval can expose several career stages at once, while a point-event index records individual observations. `IntervalRAG` uses duration-bearing macro contexts and local micro-event neighborhoods to organize the evidence around the queried period.

facts, the generator must reconstruct that temporal structure from the context (Abdallah et al., 2025; Wu et al., 2024; Chen et al., 2025b; Reichman & Heck, 2024; Fayyaz et al., 2025; Wang et al., 2025).

To address this gap, we introduce **`IntervalRAG`**, a hierarchical temporal retrieval framework that organizes evidence at two complementary levels. It extracts **Macro-Interval Units** (MIUs) for broad states or phases and **Micro-Event Units** (MEUs) for individual facts and transitions, while retaining the temporal information needed to place each unit in context. Hierarchical edges associate micro-events with their source contexts, while local temporal edges connect neighboring events and record their Allen interval relations (Allen, 1983). Given a query, retrieval first locates a relevant Macro context; it then uses the requested time to assemble nearby Micro events, boundary transitions, and source passages from that context. This separation between context localization and local evidence assembly produces a compact context in which the state and the temporal changes that explain it can be interpreted together.

We evaluate `IntervalRAG` on TimeQA, TempReason, and ComplexTR, obtaining accuracies of 60.93%, 91.79%, and 68.39%, respectively. Component ablations and retrieval analyses further examine how the representation and organization of evidence affect retrieval and answering.

Our contributions are:

- A hierarchical organization of temporal evidence that links broad narrative contexts with individual facts and transitions, while retaining the temporal information and source context needed to interpret them.
- A coarse-to-fine retrieval procedure that first locates a relevant macro context and then assembles locally ordered micro-event evidence and source passages around the requested period.
- An evaluation on three temporal QA benchmarks, together with component studies and retrieval diagnostics that separate retrieval from answer selection.

2 Related Work

Temporal RAG and Reasoning. To incorporate temporal context, recent studies have explored dynamic event trajectories, temporal graphs, and chronological evidence organization. DyG-RAG (Sun et al., 2025) indexes Dynamic Event Units; TG-RAG (Han et al., 2025) and T-GRAG (Li et al., 2025) organize time-sensitive evidence with temporal graph structures, while MRAG (Zhang et al., 2025) studies modular retrieval for time-sensitive QA. Chronological passage assembly offers a complementary way to organize retrieved text (Kim et al., 2025), and TimeR^4 (Qian et al., 2024)

connects time-aware retrieval with temporal knowledge-graph QA. Many of these approaches use discrete timestamps, event indices, or temporal filters. These signals help locate observations, but do not by themselves express interval properties such as duration, containment, and overlap. Temporal QA benchmarks such as ChronoSense (Islakoglu & Kalo, 2025) and TIME (Wei et al., 2025) further show that multiple time-scoped contexts remain difficult for language models to reconcile.

Recent interval-aware work, such as IA-RAG (Wang et al., 2026), recursively aggregates interval events into thematic forests with multi-level expansion. Our framework instead uses a two-level graph tied to source chunks: Macro-Interval Units locate a broad context, and associated Micro-Event units provide nearby evidence for the query's temporal constraints.

Graph-based RAG. Graph-based RAG introduces structured organization and global context into retrieval and generation (Peng et al., 2024). GraphRAG (Edge et al., 2024) uses community summaries for thematic retrieval; LightRAG (Guo et al., 2024) provides entity- and relation-level access; HippoRAG (Jimenez Gutierrez et al., 2024) uses associative retrieval; and E^{2}GraphRAG (Zhao et al., 2025) streamlines graph-based retrieval. These designs show how relations and abstraction levels influence evidence selection beyond isolated chunk ranking. For time-scoped questions, the temporal organization of retrieved evidence is also important because semantically related facts may belong to different periods (Sun et al., 2025). `IntervalRAG` combines interval-bearing units with a hierarchical organization so that semantic retrieval is followed by local temporal evidence selection. Node text carries factual content, hierarchical edges identify local contexts, and temporal edges expose ordering and interval relationships.

Temporal representation. Allen's algebra (Allen, 1983) and temporal annotation schemes such as TimeML (Pustejovsky et al., 2003) provide established vocabularies for interval relationships. `IntervalRAG` uses these relations to describe local evidence chronology, together with temporal features in candidate retrieval and scoring. The empirical question is how interval-bearing units affect evidence selection. We evaluate this question separately from the choice of graph-free versus graph-organized retrieval.

3 METHOD

This section presents `IntervalRAG`, a hierarchical framework for organizing and retrieving temporal evidence. Temporal questions often ask which state holds at a particular time, while the supporting evidence may be distributed across descriptions of that state and nearby changes. Its central design is to separate context localization from evidence assembly: the procedure in Figure 2 first locates a broad narrative context relevant to the question and then organizes locally continuous micro-events around the requested period. The graph records the source context, temporal extent, and local transitions of each unit; retrieval uses this structure to identify the relevant state before collecting the nearby evidence needed for answering.

3.1 HIERARCHICAL TEMPORAL GRAPH

Given a corpus $\mathcal{D}$, we build $\mathcal{G} = (\mathcal{V}^M \cup \mathcal{V}^\mu, \mathcal{E}_H \cup \mathcal{E}_T)$. Broad contexts and individual facts carry their own temporal extents, allowing retrieval to move from a relevant phase to the states and transitions that explain it. The two levels serve complementary retrieval roles: broad contexts narrow the search to a relevant phase, while individual events retain the details needed to resolve the state within it.

Node definitions. A Macro-Interval Unit (MIU) describes a broad state or phase, such as a person's affiliation with an institution. A Micro-Event Unit (MEU) describes an individual action, relation, or transition within a local narrative. Both store $v = (x, \mathcal{I}, \mathcal{N}, c)$: the extracted text x, interval $\mathcal{I} = [t_s, t_e]$, entity set $\mathcal{N}$, and source-chunk identifier c. Instantaneous events use $t_s = t_e$. Granularity is determined by the unit's content and retrieval role, not by a fixed duration threshold: a specific fact can persist for years, while a broad phase can be short. The extraction pipeline separately extracts both levels from each source chunk, and embeddings of x support semantic retrieval. Keeping both boundaries lets the retriever distinguish a state associated with the requested date from one that only starts or ends nearby; the same boundaries later determine which neighboring facts are useful

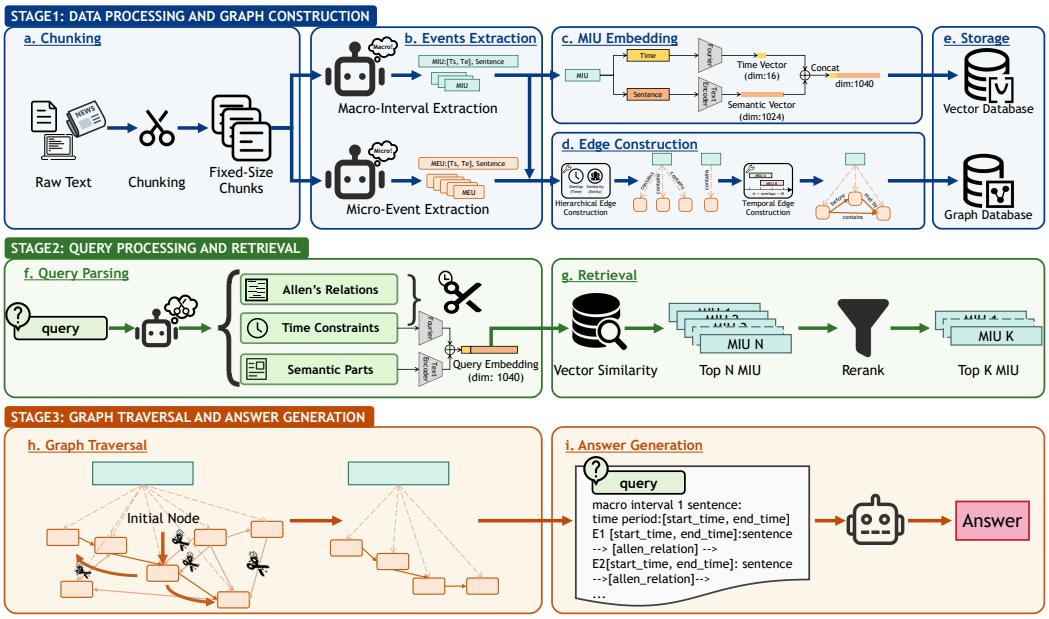


Figure 2: Overview of `IntervalRAG`. The corpus is converted into macro-interval and micro-event units connected by hierarchical and local temporal edges. A query retrieves relevant macro units, after which nearby micro-events are selected and supplied to the answer generator.

for explaining that distinction. Macro units provide the broad narrative contexts, while micro units preserve the local chronology and source-grounded details within each context.

Hierarchical edges. Each micro unit is associated with a macro context from the same source chunk. Let

$$\mathcal{M}(v^{\mu}) = \{v^{M} : c^{M} = c^{\mu},\ t_s^{M} \le t_s^{\mu} \le t_e^{M}\}. \tag{1}$$

When this set is nonempty, the construction chooses the macro with the highest entity-set overlap. Otherwise, a nearest-time fallback requires sufficient entity overlap. The selected pair forms a hierarchical edge in $\mathcal{E}_H$. This edge identifies the local retrieval context used for that micro unit; it does not require the macro interval to contain the micro unit's entire duration, and both endpoints retain their own intervals. Using a shared source chunk keeps this association grounded in local narrative context; entity and temporal compatibility then help select the macro context used for local retrieval. Appendix D specifies the assignment and missing-time cases.

Local temporal edges. Within each macro context, micro units with parseable times are sorted by start time. For each unit, we connect nearby predecessors and successors, with at most four neighbor selections per unit. For a selected pair (v_i^{μ}, v_j^{μ}), the edge stores the Allen relation $r_{ij} = \mathrm{Allen}(\mathcal{I}_i^{\mu}, \mathcal{I}_j^{\mu})$. Thus, $\mathcal{E}_T$ stores a local temporal neighborhood rather than every pairwise interval relation. These edges make nearby transitions accessible from a retrieved fact, including transitions that are weak semantic matches to the question but important for interpreting the local chronology. The hierarchical edge keeps that expansion within the selected context, while node text preserves the factual content.

3.2 Coarse-to-Fine Retrieval under Temporal Constraints

Query representation. We parse a question into

$$q \longrightarrow (\mathbf{e}_q, \mathcal{I}_q, r_q, \mathcal{N}_q), \tag{2}$$

where $\mathbf{e}_q$ represents the semantic query, $\mathcal{I}_q$ its time bounds, r_q its temporal relation, and $\mathcal{N}_q$ its entities. Keeping these fields separate lets retrieval assess topical relevance, target entities, and temporal scope as distinct constraints rather than relying on a single undifferentiated similarity match. For event-relative questions without an explicit date, an initial retrieval step resolves the anchor event before temporal retrieval (Appendix G).

Macro context retrieval. Macro units provide the entry points into the graph because they describe candidate states at the scale of a narrative phase. They locate the broad context before finer-grained evidence is selected. Their index combines semantic text features with Fourier features of the start and end boundaries. Query-boundary searches produce candidate lists that are merged by macro ID. For query and macro index vectors $\mathbf{z}_q, \mathbf{z}_m$, the vector score is

$$S_{\mathrm{vec}}(q, m) = \frac{\mathbf{z}_q \cdot \mathbf{z}_m}{\|\mathbf{z}_q\|\|\mathbf{z}_m\|}. \tag{3}$$

Encoding both boundaries gives initial recall access to the candidate's temporal extent alongside its meaning, so a long-running state can be retrieved even when its nearest textual mention is not dated at the query point. The temporal feature weight controls this initial recall and is distinct from the temporal coefficient used next.

Reranking. We select the macro set $\mathcal{A}$ from the merged candidates using

$$S_{\mathrm{rr}}(q, v^M) = \alpha \,\mathrm{sim}_{\mathrm{sem}}(q, x^M) + \beta \,\mathrm{sim}_{\mathrm{temp}}(\mathcal{I}_q, \mathcal{I}^M) + \gamma \,\mathrm{sim}_{\mathrm{ent}}(\mathcal{N}_q, \mathcal{N}^M). \tag{4}$$

The terms measure cross-encoder relevance, interval proximity, and query-entity coverage. These signals address complementary ambiguities: semantic relevance can identify the right topic but a different period, while temporal proximity alone can favor an event about the wrong entity. The interval term rewards overlap as well as proximity, so a long-running state can remain relevant to a date well after its start. The top-K macro units by this score define the local micro-event candidate set $\mathcal{C} = \{v^\mu : \exists v^M \in \mathcal{A}, (v^M, v^\mu) \in \mathcal{E}_H\}$. The selected macro units identify both a relevant topic and the temporal context in which nearby micro-event evidence should be organized. This separation keeps the initial search focused on finding the right context, while the subsequent traversal can preserve local continuity around the requested period.

3.3 Temporal Graph Traversal and Answer Generation

The answer to a state-at-time question may depend on events that mark its beginning or end, even when the requested date lies between them. The selected macro contexts narrow the search to a relevant narrative, and local traversal retrieves these nearby transitions rather than relying on the macro description alone.

Seed selection. For each selected macro, the search begins at a micro event whose start time is closest to the query's temporal reference. For a bounded query, this reference is the interval midpoint $t_q^\star$:

$$v_s^\mu = \arg \min_{v^\mu \in \mathcal{C}(v^M)} |t_s^\mu - t_q^\star|. \tag{5}$$

The parsed query relation determines the temporal search window for local selection. The implementation uses a recall buffer around its boundaries; the configured value is given in Appendix C.

Local traversal. From each seed, the algorithm follows local temporal neighbors toward earlier and later start times, subject to a depth budget, to assemble a locally continuous evidence sequence. A step is temporally relevant when the neighboring event, or the span connecting it to the current event, intersects the search window. The span test brings in transitions on either side of the queried period, which can supply the beginning or end of a state described elsewhere in the local narrative. The retrieved facts retain their extracted intervals; persistence between them is assessed from their text during answering. Search stops after the budget or a boundary is reached. Appendix D gives the search-state and stopping rules.

Context construction. Selected events are deduplicated and ordered by time, with their temporal extents and the relations between successive events exposed in the textual context. A separate chunk budget supplies source passages selected from the reranked macro units. The generator receives the question, the locally organized event context, and source text and selects an answer under the question's temporal conditions. Presenting events in temporal order makes their relation to the queried period explicit, instead of leaving the generator to reconstruct chronology from separate facts. For a tenure question, the resulting context can present the affiliation together with evidence of its appointment and departure, allowing the answer to be grounded in the requested part of that tenure.

Table 1: Overall performance on three temporal QA benchmarks. The best and second-best available values in each dataset–metric column are shown in **bold** and underlined, respectively. A dash indicates that the corresponding reference did not report that metric under the same protocol.

Method	**TimeQA** Chen et al. (2021) (↑)		**TempReason** Tan et al. (2023) (↑)		**ComplexTR** Tan et al. (2024) (↑)	
	Acc. (%)	Recall (%)	Acc. (%)	Recall (%)	Acc. (%)	Recall (%)
Vanilla RAG Lewis et al. (2020)	37.58	44.14	52.01	64.38	42.55	51.46
GraphRAG-L Edge et al. (2024)	40.26	46.28	56.11	67.55	43.16	54.29
GraphRAG-G Edge et al. (2024)	10.10	13.83	8.81	11.34	20.97	30.81
LightRAG-L Guo et al. (2024)	34.94	39.89	54.23	66.82	41.03	49.46
LightRAG-G Guo et al. (2024)	6.66	8.12	13.54	15.27	12.46	13.56
LightRAG-H Guo et al. (2024)	36.36	43.06	50.01	62.95	42.68	53.59
HippoRAG Jimenez Gutierrez et al. (2024)	39.99	45.39	69.80	80.54	44.68	55.28
E^2GraphRAG Zhao et al. (2025)	40.48	50.19	61.29	73.58	38.29	54.99
TG-RAG (Han et al., 2025)	44.74	56.09	35.37	38.47	51.98	63.54
T-GRAG (Li et al., 2025)	23.34	34.21	54.86	65.32	28.57	34.78
DyG-RAG Sun et al. (2025)	58.78	67.02	<u>84.75</u>	<u>91.47</u>	55.62	69.88
IA-RAG Wang et al. (2026)	**61.72**	<u>69.25</u>	80.21	89.64	<u>65.95</u>	<u>77.42</u>
`IntervalRAG` (ours)	<u>60.93</u>	**69.39**	**91.79**	**95.49**	**68.39**	**79.28**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate `IntervalRAG` on three temporal QA benchmarks that target distinct facets of temporal reasoning Sun et al. (2025): (i) **TimeQA** Chen et al. (2021) for implicit temporal inference over long documents; (ii) **TempReason** Tan et al. (2023) for fine-grained event-state grounding; and (iii) **ComplexTR** Tan et al. (2024) for complex multi-hop temporal reasoning.

Baselines. We compare `IntervalRAG` against three categories of RAG frameworks: (i) **Standard RAG**, represented by **Vanilla RAG** Lewis et al. (2020), which performs semantic vector retrieval over text chunks; (ii) **Graph-based RAG**, including **GraphRAG** Edge et al. (2024) (hierarchical community summaries), **LightRAG** Guo et al. (2024) (dual-level indexing), **HippoRAG** Jimenez Gutierrez et al. (2024) (associative retrieval), and **E^2GraphRAG** Zhao et al. (2025) (efficient pipeline retrieval); and (iii) **Temporal RAG**, represented by **DyG-RAG** Sun et al. (2025), an event-centric framework that uses Dynamic Event Units to track temporal dynamics. For variants, GraphRAG-L/G denote local/global search, and LightRAG-L/G/H denote low-level, high-level, and hybrid retrieval, respectively. We additionally include **TG-RAG** (Han et al., 2025), **T-GRAG** (Li et al., 2025), and **IA-RAG** (Wang et al., 2026); Appendix E describes their retrieval procedures and benchmark comparisons. Global-search modes provide configuration comparisons; DyG-RAG and IA-RAG are the closest temporal retrieval comparisons.

Implementation details. `IntervalRAG` uses **Qwen2.5-14B-Instruct** Qwen et al. (2025) for event extraction and answer generation, **BGE-M3** Chen et al. (2025a) for text encoding, and **dslim/bert-base-NER** Devlin et al. (2019) for named-entity recognition. We use the benchmark corpora and baseline reference results from the DyG-RAG evaluation protocol Sun et al. (2025). Appendix C gives the implementation configuration.

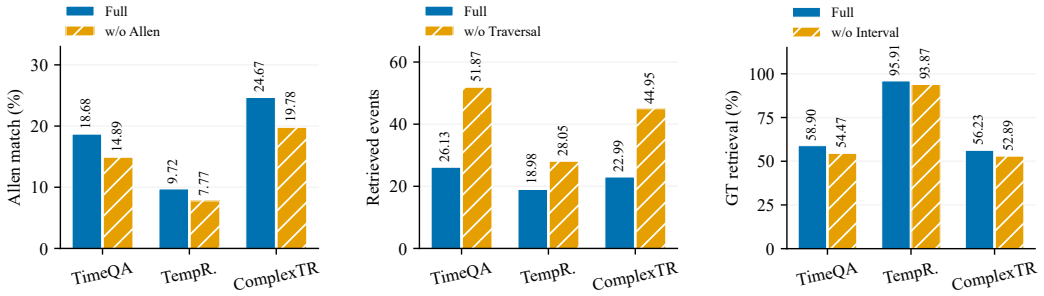
Evaluation. We evaluate all 2,613, 5,397, and 329 questions in TimeQA, TempReason, and ComplexTR, including empty-context cases. Acc. is defined in Eq. equation 8. Recall measures token overlap for single-answer questions and component coverage for multi-answer questions. Appendix H gives the exact parsing rules and reports final-line F1/EM diagnostics separately from the main task scores.

4.2 MAIN RESULTS

Table 1 compares end-to-end question-answering performance. The ablations below and retrieval diagnostics in Appendix I examine how evidence organization contributes to these results.

Table 2: **Bold** indicates the best score within each column.

Variant	**TimeQA** Chen et al. (2021)		**TempReason** Tan et al. (2023)		**ComplexTR** Tan et al. (2024)	
	Acc. (%) ↑	Recall (%) ↑	Acc. (%) ↑	Recall (%) ↑	Acc. (%) ↑	Recall (%) ↑
`IntervalRAG`	**60.93**	**69.39**	**91.79**	**95.49**	**68.39**	**79.28**
w/o Allen	57.60	66.72	90.95	94.73	66.57	77.52
w/o Traversal	58.60	66.96	90.30	94.17	66.26	76.27
w/o Interval	55.46	64.66	89.40	93.52	65.65	75.07



(a) Allen Relation Match Rate (b) Avg. retrieved events (c) GT containment rate

Figure 3: Analysis metrics for `IntervalRAG`. (a) Query-level Allen-relation match rate with and without relation guidance. (b) Mean event count after local traversal or top-K retrieval. (c) Ground-truth interval containment rate with interval or timestamp representations.

Overall performance. `IntervalRAG` obtains accuracy (Acc.) of 60.93%, 91.79%, and 68.39%, corresponding to $1{,}592/2{,}613$, $4{,}954/5{,}397$, and $225/329$ questions. Recall is reported alongside accuracy, with all required answer components counted. The table reports the corresponding Acc./Recall values for the listed temporal methods.

Evidence availability in multi-hop questions. ComplexTR shows the largest accuracy difference. Its input contains all gold answer components for $263/329$ questions (79.94%), compared with 225 hits. This gap motivates inspecting both evidence selection and generation: a useful temporal context still requires the model to select the correct subject, relation, and interval.

State localization on TempReason. `IntervalRAG` achieves 91.79% accuracy on TempReason, where questions require identifying a state or role at a particular time. The retrieved inputs contain all answer components for 97.81% of questions. The remaining gap highlights the need to resolve competing subjects, roles, and tenure boundaries even when relevant evidence is available.

4.3 Ablation Study

Table 2 examines interval representation, traversal, and relation guidance, while Figure 3 shows the associated analyses. Each variant is compared with its corresponding full-system control. We additionally compare with graph-free raw-chunk controls (Table 3). The variants *w/o Allen*, *w/o Traversal*, and *w/o Interval* respectively remove Allen-relation guidance, replace graph traversal with top-K retrieval, and replace interval representations with timestamps. The variants use the corresponding control settings apart from the component being removed.

Effect of Allen relations. Removing Allen-relation guidance lowers TimeQA accuracy by 3.33 percentage points and the query-level relation-match rate from 18.68% to 14.89% (Figure 3a).

Effect of graph traversal. Replacing local traversal with top-K event retrieval reduces ComplexTR accuracy by 2.13 percentage points. On TimeQA, the mean retrieved event count increases from 26.13 to 51.87 (Figure 3b) while accuracy decreases. More retrieved events do not necessarily provide more useful evidence. This variant retains the event graph and differs from the graph-free controls below.

Table 4: Accuracy (%) with different Qwen2.5-Instruct generator sizes. Avg. Acc. is the unweighted mean across the three datasets.

Generator	Method	TimeQA	TempReason	ComplexTR	Avg. Acc. ↑
3B	DyG-RAG	40.07	61.15	37.99	46.40
	IntervalRAG	**46.35**	**84.82**	**48.94**	**60.04**
7B	DyG-RAG	46.80	61.33	45.29	51.14
	IntervalRAG	**57.33**	**90.62**	**61.09**	**69.68**
14B	DyG-RAG	58.94	82.38	59.57	66.96
	IntervalRAG	**60.93**	**91.79**	**68.39**	**73.70**

Interval modeling vs. timestamps. Replacing intervals with timestamps lowers accuracy by 5.47, 2.39, and 2.74 percentage points across the three datasets. The loss is largest among the three component removals in Table 2 on each dataset, supporting the role of duration-bearing evidence in the retrieval procedure.

Graph-free retrieval (NoGraph). The graph-free pipeline retrieves raw 1,200-token chunks using BGE-M3 and supplies them to Qwen2.5-14B-Instruct. Semantic retrieval ranks chunks directly, while hard interval filtering applies the query-time condition to recalled chunks. These variants remove event-graph construction, macro–micro expansion, traversal, and event-timeline serialization. Thus, they compare evidence pipelines, whereas the traversal ablation in Table 2 retains the event representation.

Table 3: Graph-free raw-chunk controls: accuracy (Acc., %). k counts raw chunks.

Retrieval	Raw chunks k	TimeQA	TempReason	ComplexTR
Semantic	1	32.41	71.04	49.54
Semantic	2	48.37	79.08	57.14
Semantic	5	58.94	78.30	52.89
Hard interval filter	1	36.43	71.73	47.11
Hard interval filter	2	48.72	78.75	54.10
Hard interval filter	5	52.97	77.78	48.94
IntervalRAG (chunk=2)	2	60.93	91.79	68.39

With two source chunks (Table 3), IntervalRAG reaches 60.93%, 91.79%, and 68.39% on TimeQA, TempReason, and ComplexTR. Relative to semantic raw-chunk retrieval with $k = 2$, the accuracy gains are 12.56, 12.71, and 11.25 percentage points; relative to hard interval filtering with $k = 2$, they are 12.21, 13.04, and 14.29 points. All three settings supply two source chunks, and IntervalRAG additionally supplies its retrieved event context.

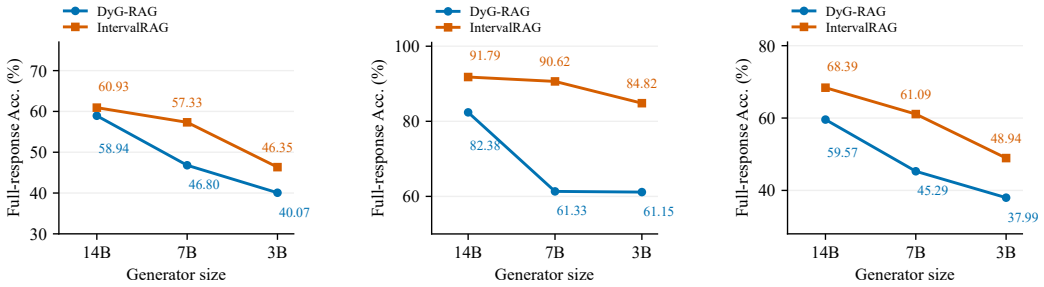
Chunk-count trends are evaluated separately within each graph-free retrieval method. Within semantic retrieval, increasing k from 1 to 2 to 5 raises TimeQA accuracy from 32.41% to 48.37% to 58.94%; TempReason and ComplexTR peak at $k = 2$ (79.08% and 57.14%) before declining at $k = 5$ (78.30% and 52.89%). Within hard interval filtering, TimeQA likewise improves from 36.43% to 48.72% to 52.97%, while TempReason and ComplexTR peak at $k = 2$ (78.75% and 54.10%) and decline at $k = 5$ (77.78% and 48.94%).

4.4 Additional Analysis

We analyze IntervalRAG's answering performance with different generator sizes and across query–evidence temporal-distance groups. Appendix I reports stage-wise retrieval coverage and final-context answer availability.

Results with different generator sizes. Table 4 reports accuracy with 3B, 7B, and 14B Qwen2.5-Instruct generators. Figure 4 visualizes the per-dataset results.

The 7B IntervalRAG results reach 57.33%, 90.62%, and 61.09% on TimeQA, TempReason, and ComplexTR. With the 3B generator, the corresponding scores are 46.35%, 84.82%, and 48.94%.



(a) TimeQA (b) TempReason (c) ComplexTR

Figure 4: Accuracy with 14B, 7B, and 3B Qwen2.5-Instruct generators on the three temporal QA benchmarks.

These measurements show that the system remains usable with smaller generators, with particularly high answer inclusion on TempReason.

Query–evidence temporal distance. We partition questions by their estimated query–evidence temporal distance into Near, Middle, and Far groups. Group membership is fixed across methods and independent of model predictions. Questions without an estimated distance are distributed across groups (198 TimeQA, 4 TempReason, and 3 ComplexTR questions). This analysis measures performance across temporal-distance strata.

Table 5: Accuracy across query–evidence distance groups. Group membership is identical across methods: TimeQA 871/871/871, TempReason 1799/1799/1799, and ComplexTR 110/110/109.

Dataset	Method	Near	Middle	Far
TimeQA	DyG-RAG	**57.86**	66.02	52.93
TimeQA	`IntervalRAG`	57.41	**70.95**	**54.42**
TempReason	DyG-RAG	80.32	85.44	81.38
TempReason	`IntervalRAG`	**91.66**	**93.33**	**90.38**
ComplexTR	DyG-RAG	60.00	68.18	50.46
ComplexTR	`IntervalRAG`	**64.55**	**78.18**	**62.39**

Table 5 shows that accuracy is highest in the Middle group on each dataset rather than changing monotonically with distance. IntervalRAG exceeds DyG-RAG in the Middle and Far groups on all three datasets, and in the Near group on TempReason and ComplexTR; TimeQA Near is slightly lower (57.41% versus 57.86%). These patterns describe performance across the groups rather than a causal effect of temporal distance.

Efficiency analysis. We additionally evaluate indexing and query-time efficiency, with the full results reported in Appendix L.

5 Conclusion

We presented `IntervalRAG`, a temporal RAG framework that retrieves facts together with evidence about when they hold. Its hierarchical interval graph uses macro states and phases to locate a context that covers the query, then gathers micro facts and neighboring transitions that explain the context's boundaries. In this way, temporal extent is carried from the indexed units into candidate selection, local evidence assembly, and the chronology supplied to generation, rather than being treated as a final post-retrieval check. `IntervalRAG` reaches 60.93%, 91.79%, and 68.39% accuracy on TimeQA, TempReason, and ComplexTR, respectively. Interval removal causes the largest accuracy loss among the three component ablations in Table 2, while retrieval diagnostics reveal remaining losses between retrieving an answer-bearing candidate and selecting the answer from it. Uncertain extracted boundaries and competing temporal evidence remain limitations. Better calibration of uncertain intervals is a useful direction for improving evidence organization.

AI USE STATEMENT

Generative AI tools assisted with language editing, manuscript organization, and consistency checks. Large language models are also used for temporal evidence extraction, query interpretation, and answer generation in IntervalRAG, and for the extraction-quality audit, as described in the method and appendix. The authors are responsible for the final manuscript, scientific claims, and reported results.

REFERENCES

Abdelrahman Abdallah, Bhawna Piryani, Jonas Wallat, Avishek Anand, and Adam Jatowt. Tempretriever: Fusion-based temporal dense passage retrieval for time-sensitive questions. *arXiv preprint arXiv:2502.21024*, 2025.

James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2025a. URL https://arxiv.org/abs/2402.03216.

Wenhu Chen, Xinyi Wang, and William Yang Wang. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*, 2021.

Ziyang Chen, Erxue Min, Xiang Zhao, Yunxin Li, Xin Jia, Jinzhi Liao, Jichao Li, Shuaiqiang Wang, Baotian Hu, and Dawei Yin. a question answering dataset for temporal-sensitive retrieval-augmented generation. *Scientific Data*, 12(1):1855, 2025b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9136–9152, 2025.

Anoushka Gade, Jorjeta G Jetcheva, and Hardi Trivedi. It's about time: Incorporating temporality in retrieval augmented language models. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pp. 75–82. IEEE, 2025.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.

Jiale Han, Austin Cheung, Yubai Wei, Zheng Yu, Xusheng Wang, Bing Zhu, and Yi Yang. Rag meets temporal graphs: Time-sensitive modeling and retrieval for evolving knowledge, 2025. URL https://arxiv.org/abs/2510.13590.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55, January 2025. ISSN 1558-2868. doi: 10.1145/3703155. URL http://dx.doi.org/10.1145/3703155.

Duygu Sezen Islakoglu and Jan-Christoph Kalo. Chronosense: Exploring temporal understanding in large language models with time intervals of events. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 590–602, 2025.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569, 2024.

Byeongjeong Kim, Jeonghyun Park, Joonho Yang, and Hwanhee Lee. Chronological passage assembling in rag framework for temporal question answering, 2025. URL https://arxiv.org/abs/2508.18748.

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.

Dong Li, Yichen Niu, Ying Ai, Xiang Zou, Biqing Qi, and Jianxing Liu. T-grag: A dynamic graphrag framework for resolving temporal conflicts and redundancy in knowledge retrieval. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 11880–11889, 2025.

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

Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. It's high time: A survey of temporal question answering, 2025. URL https://arxiv.org/abs/2505.20243.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34, 2003.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. Timer4: Time-aware retrieval-augmented large language models for temporal knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6942–6952, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Benjamin Reichman and Larry Heck. Dense passage retrieval: Is it retrieving? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13540–13553, 2024.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17, 2023.

Qingyun Sun, Jiaqi Yuan, Shan He, Xiao Guan, Haonan Yuan, Xingcheng Fu, Jianxin Li, and Philip S Yu. Dyg-rag: Dynamic graph retrieval-augmented generation with event-centric reasoning. *arXiv preprint arXiv:2507.13396*, 2025.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*, 2023.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards robust temporal reasoning of large language models via a multi-hop qa dataset and pseudo-instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6272–6286, 2024.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence, 2025. URL https://arxiv.org/abs/2504.13079.

Xiaoman Wang, Yaoze Zhang, Wenzhuo Fan, Hongwei Zhang, Ding Wang, Guohang Yan, Song Mao, Botian Shi, Yunshi Lan, and Pinlong Cai. Ia-rag: Interval-algebra-driven temporal reasoning for dynamic knowledge retrieval. *arXiv preprint arXiv:2606.06044*, 2026.

Shaohang Wei, Wei Li, Feifan Song, Wen Luo, Tianyi Zhuang, Haochen Tan, Zhijiang Guo, and Houfeng Wang. Time: A multi-level benchmark for temporal reasoning of llms in real-world scenarios. *arXiv preprint arXiv:2505.12891*, 2025.

Feifan Wu, Lingyuan Liu, Wentao He, Ziqi Liu, Zhiqiang Zhang, Haofen Wang, and Meng Wang. Time-sensitve retrieval-augmented generation for question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2544–2553, 2024.

Siyue Zhang, Yuxiang Xue, Yiming Zhang, Xiaobao Wu, Anh Tuan Luu, and Chen Zhao. Mrag: A modular retrieval framework for time-sensitive question answering. In *EMNLP (Findings)*, pp. 3080–3118, 2025.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. E^2GraphRAG: Streamlining graph-based rag for high efficiency and effectiveness, 2025. URL https://arxiv.org/abs/2505.24226.

Zulun Zhu, Haoyu Liu, Mengke He, and Siqiang Luo. Right answer at the right time - temporal retrieval-augmented generation via graph summarization, 2025. URL https://arxiv.org/abs/2510.16715.

Overview of the Appendix

This appendix is organized as follows.

- Appendix A: the three temporal QA benchmarks (TimeQA, TempReason, ComplexTR) and how we adapt them to RAG.
- Appendix B: Allen's interval relations used to label temporal evidence.
- Appendix C: hardware, model versions, and retrieval/traversal hyperparameters.
- Appendix D: operational construction, local traversal, and context assembly.
- Appendix E: additional temporal retrieval methods and design comparison.
- Appendix F: vector-database construction and query embedding.
- Appendix G: query parsing and event-anchored recovery for queries without explicit times.
- Appendix H: inclusion metrics and short-answer F1/EM diagnostics.
- Appendix I: stage-wise retrieval coverage and final-context answer availability.
- Appendix J: an LLM-as-a-judge audit of the event extractor on ComplexTR.
- Appendix K: a stage-wise decomposition of `IntervalRAG`'s remaining errors.
- Appendix L: offline indexing and online query-time efficiency.
- Appendix M: three end-to-end qualitative case studies, one per benchmark.
- Appendix N: the macro-interval, micro-event, and answer-generation prompts.
- Appendix O: limitations of `IntervalRAG` and concrete future-work directions.
- Appendix P: declaration of LLM usage in this work.

A Datasets

We evaluate `IntervalRAG` on three established temporal QA benchmarks: **TimeQA** Chen et al. (2021), **TempReason** Tan et al. (2023), and **ComplexTR** Tan et al. (2024). They cover implicit temporal inference, event-state localization, and multi-hop temporal reasoning over Wikipedia-derived corpora. We follow the DyG-RAG-style adaptation of these datasets to RAG Sun et al. (2025).

A summary of corpus statistics is given in Table 6; per-dataset descriptions follow.

Table 6: Corpus statistics for the three temporal QA benchmarks used in our experiments.

Dataset	#Docs	#Tokens	Avg. doc. length	#Questions
TimeQA Chen et al. (2021)	3,159	1.81M	576 tokens	2,613
TempReason Tan et al. (2023)	2,721	2.67M	980 tokens	5,397
ComplexTR Tan et al. (2024)	112	0.24M	2,144 tokens	329

TimeQA. TimeQA targets implicit temporal reasoning: the system must infer event time ranges that are not explicitly annotated, going beyond keyword matching or timestamp lookup to perform contextual and cross-sentence reasoning. Successful systems must organize timelines and integrate temporally distributed information across long contexts.

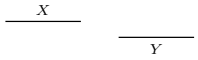
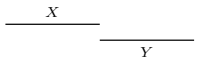
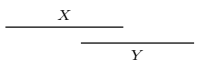
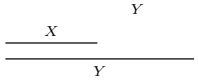
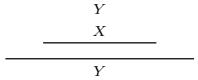
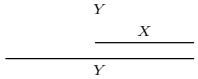
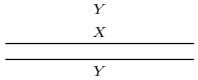
TempReason. TempReason focuses on event-state localization: the system must determine an entity's state or role at a specific point in time (e.g., whether a person held a particular position on a given date). It is the largest of the three benchmarks and is well suited for measuring fine-grained temporal modeling and precise time-aware retrieval, particularly for long-duration states and state transitions.

ComplexTR. ComplexTR is designed for multi-hop temporal reasoning: each question requires linking multiple temporally separated yet semantically related events to form a complete temporal logic chain. Although its corpus is small, every document is long and every question is deliberately difficult, jointly stressing long-range temporal dependencies and accurate event-sequence organization.

B Allen's Interval Algebra

Allen's interval algebra Allen (1983) describes the relative boundaries of two intervals. For non-degenerate intervals, its 13 mutually exclusive relations comprise six asymmetric pairs and one self-inverse identity relation. Table 7 summarizes these relations, which annotate temporal edges and the chronology presented to the generator. Query relations determine the local search window; the endpoint and span tests used during traversal are specified in Appendix D.

Table 7: Summary of Allen's interval algebra. The inverse column covers the corresponding reverse relations; equality is self-inverse.

Relation	Inverse	Sketch	Meaning
$X < Y$	$Y > X$		X ends before Y starts.
$X\ m\ Y$	$Y\ mi\ X$		Y starts exactly when X ends.
$X\ o\ Y$	$Y\ oi\ X$		X and Y overlap on a non-empty interval.
$X\ s\ Y$	$Y\ si\ X$		X starts with Y and ends earlier.
$X\ d\ Y$	$Y\ di\ X$		X is fully contained in Y.
$X\ f\ Y$	$Y\ fi\ X$		X ends with Y and starts later.
$X = Y$	self		X and Y have identical boundaries.

C Implementation Details and Hyperparameters

This appendix complements Section 4.1 with the model backbones and the implementation configuration.

Models and configuration. We use `Qwen2.5-14B-Instruct` for event extraction and answer generation, `BGE-M3` for semantic encoding, and `dslim/bert-base-NER` for named-entity recognition. Documents are split into 1,200-token chunks. All experiments run on a server with eight RTX 3090 GPUs. Table 8 summarizes the implementation configuration.

Retrieval configuration. The context budget is 20 reranked seeds and two source chunks. The remaining retrieval and traversal parameters are listed below.

Table 8: Implementation settings. The vector time weight and reranking time coefficient are separate parameters.

Setting	Value
Vector candidate setting (`topk1`)	2,000
Vector `time_weight`	0.1
Rerank seeds (K)	20
Final chunk budget	2
Rerank semantic/entity/time weights	0.4 / 0.4 / 0.2
Temporal decay rate	0.02
Walk depth/nodes/branches	5 / 5 / 3
Retrieval buffer (days)	1,825
Generation concurrency	32

D Operational Details of Evidence Construction and Retrieval

This section expands the graph and retrieval operations in Section 3. The two levels serve different purposes: macro text locates a broad context, while micro text preserves the individual facts gathered within that context.

Extracted records and provenance. The source corpus is segmented into chunks, and separate extraction passes obtain macro contexts and micro facts from each chunk. Both record types retain their text, temporal fields, entities, and source-chunk provenance; node identifiers are generated by the indexing code. The macro level is extracted directly from the narrative rather than recursively summarized from the micro graph.

Macro–micro assignment. Each Micro is associated with a Macro from the same source chunk using temporal compatibility together with semantic and entity support. The selected pair forms a hierarchical edge, while weak or unresolved cases remain explicitly marked for auditing rather than being silently discarded. This association identifies a local context; it does not assert strict containment of every part of a Micro interval.

Temporal edge construction. Within each Macro context, Micro units with usable times are ordered chronologically and connected to a small set of nearby predecessors and successors. Each selected pair stores its interval relation and edge weight, yielding a local temporal neighborhood rather than a complete pairwise Allen graph.

Initial recall and reranking. The Macro index queries the available temporal boundaries and merges the returned lists by Macro ID before reranking. Reranking combines semantic, temporal, and entity signals; the top-K Macros provide the anchors for local Micro retrieval and source-chunk selection.

Local search state. For each anchor Macro, the algorithm selects a nearby Micro seed and expands in both temporal directions through the stored neighborhood under a depth budget. The query relation defines the search window; neighboring intervals and the span between adjacent events provide the relevance test. A bounded buffer and a Macro-level fallback preserve useful context when the local neighborhood does not directly intersect the query window.

Allen labels annotate the assembled chronology, while the traversal remains a bounded evidence-selection procedure rather than a complete Allen-logic execution. The search buffer affects which evidence is considered; it does not rewrite the intervals stored on the nodes.

Source passages and serialization. Each reranked macro contributes a vote to its source chunk. Chunks are ranked by vote count, with the earliest macro rank breaking ties, and the configured chunk budget determines how many passages are supplied. Passage selection therefore starts from the reranked macro set and does not require the source of every traversed micro-event to be included. Event evidence and selected passages can consequently provide complementary information. Micro records are deduplicated and ordered by time. Interval relations between successive serialized records can be recomputed from their endpoints even if that pair is not a stored traversal edge. The generation context contains event text, temporal annotations, and the selected passages; the answer template is given in Appendix N.

Scope of local evidence. An interval-spanning step retrieves neighboring facts that can help interpret a state between transitions. It is not, by itself, a proof that a state persisted throughout the gap. The generator must still resolve the subject, relation, and temporal conditions from the supplied text. This distinction motivates measuring both context answer coverage and the final selected answer.

E Baseline Descriptions

Vanilla RAG. Vanilla RAG (Lewis et al., 2020) retrieves the most similar source chunks with a standard dense retriever and supplies them directly to the generator, without an explicit graph or temporal index.

GraphRAG. GraphRAG (Edge et al., 2024) builds a graph from extracted entities and relations and supports both local entity-centered retrieval and global retrieval through community summaries. It provides a graph-structured comparison without explicitly modeling temporal validity intervals.

LightRAG. LightRAG (Guo et al., 2024) maintains low-level entity/relation representations together with higher-level summaries and supports low-level, high-level, and hybrid retrieval. Its layered access pattern offers a comparison for separating coarse context selection from local evidence retrieval.

HippoRAG. HippoRAG (Jimenez Gutierrez et al., 2024) uses an entity-centered knowledge graph and associative retrieval to connect query concepts with supporting passages. The graph is used as a memory structure, while temporal validity is not an explicit indexing dimension.

E^{2}GraphRAG. E^{2}GraphRAG (Zhao et al., 2025) streamlines graph construction and query-time graph traversal for retrieval-augmented generation. We include it as an efficiency-oriented graph baseline alongside the more summary- or entity-centric systems.

DyG-RAG. DyG-RAG (Sun et al., 2025) represents changing facts as dynamic event units and retrieves temporal evidence from an event graph. Its benchmark adaptation supplies the reference corpus processing and evaluation protocol used for the comparisons in this paper.

TG-RAG. TG-RAG (Han et al., 2025) organizes temporal evidence through a temporal graph and time-aware retrieval. It uses the common question splits, generator family, and evaluator, with BGE-M3 as the retrieval backbone. Its temporal index provides a comparison to the macro-anchored interval neighborhoods used by `IntervalRAG`.

T-GRAG. T-GRAG (Li et al., 2025) retrieves local temporal subgraphs and entity paths from timestamped document snapshots. Its temporal organization is based on document snapshots, whereas `IntervalRAG` extracts the validity interval of individual facts within a narrative.

IA-RAG. IA-RAG (Wang et al., 2026) organizes interval events into a recursively summarized Thematic Forest and uses query-time boundary refinement with relation-guided expansion. `IntervalRAG` instead retrieves source-grounded Macro-Interval Units first, then organizes their associated Micro-Events into local temporal evidence around the queried state.

F Vector Database and Query Embedding

The standard retrieval path embeds Macro-Interval Units (MIUs) using semantic content and temporal boundaries, then accesses their associated Micro-Event Units (MEUs) through hierarchical edges. A separate micro index supports event-anchor recovery when a query lacks an explicit date. Each macro interval contributes start- and end-boundary index entries so that interval-aware retrieval can match either boundary.

The vector embedding of the i-th unit is the concatenation of a semantic part and a temporal part:

$$z_i \;=\; \mathrm{Concat}\Big(h^{(i)}_{\mathrm{sem}},\, h^{(i)}_{\mathrm{time}}\Big)\,, \tag{6}$$

where $h^{(i)}_{\mathrm{sem}}$ is the semantic embedding of the unit and $h^{(i)}_{\mathrm{time}}$ is its time embedding produced by a Fourier time encoder:

$$h^{(i)}_{\mathrm{time}} \;=\; \phi(t_i), \tag{7}$$

with t_i denoting the start (or end) timestamp of the unit. A query-boundary search uses the same composite scheme: the query semantic vector $\mathbf{e}_q$ (Section 3.2) is concatenated with the Fourier features of that boundary. When the query boundaries differ, both searches contribute to the candidate union.

G Query Parsing and Recovery

Query decomposition. Each query q is parsed into $(\mathbf{e}_q, \mathcal{I}_q, r_q, \mathcal{N}_q)$ as in Eq. equation 2. The semantic component captures the question's intent, while $\mathcal{N}_q$ collects its entities. The time bounds and relation specify the requested period, which is used in macro scoring and to construct the micro search window. For example:

- For the *during* relation, the query interval is bounded: $\mathcal{I}_q = [t_a, t_b]$. For instance, "during 1900 to 1950" yields $[1900, 1950]$.
- For the *after* relation, the query interval is right-open: $\mathcal{I}_q = [t_a, +\infty)$. For instance, "after 2000" yields $[2000, +\infty)$.

Query recovery for missing explicit times. Query parsing is performed independently from corpus-side temporal extraction. When the query contains an explicit time anchor (a year, date, or time span), `IntervalRAG` maps it directly to a query interval and Allen relation; for example, "two years before 2020" is converted into the corresponding target interval before retrieval. When the query carries no explicit computable time but instead refers to an anchor event, `IntervalRAG` applies an event-anchored recovery procedure:

1. identify the anchor event and its associated entities in the query;
2. retrieve evidence for the anchor event and read off its interval;
3. apply the relative relation or offset (*before*, *after*, "N years/months earlier", etc.) to derive the target interval;
4. rewrite the query with the recovered interval and re-enter the standard interval-aware retrieval pipeline.

For example, when a query asks who held a position before a person took office but does not provide the inauguration date, `IntervalRAG` first retrieves the inauguration event, extracts its time, rewrites the query with the recovered interval, and then proceeds with macro-anchoring and micro-event filtering.

Query embedding. The parsed query is embedded into a dense vector that fuses semantic and temporal information following the same scheme as Eq. equation 6: the semantic intent is encoded by the text encoder, and the parsed query interval is processed by the Fourier time encoder used at index time. This consistency between query-side and index-side encoding is essential for the cosine-similarity score in Eq. equation 3 to be well calibrated.

H Evaluation Metrics

Accuracy. Let $\mathcal{A}_q$ be the nonempty required gold components and y_q the complete response. Gold strings are split using the evaluation protocol's pipe, comma, and quoted-list delimiters. Then

$$\mathrm{Acc}_{\mathrm{full}} = \frac{100}{|Q|} \sum_{q \in Q} \mathbb{I}[\forall a \in \mathcal{A}_q : \ a \text{ occurs in } \mathrm{norm}(y_q)] \, . \tag{8}$$

All required components must occur. The metric measures answer inclusion in the response. Empty-context and unsuccessful records remain in the evaluation denominator.

Recall. For one required answer, Recall is token multiset overlap divided by the number of gold tokens. For multiple required components, it is the fraction whose normalized strings occur in the response. The dataset score averages these per-question values. This inherited multi-answer rule is component coverage, not token recall.

Short-answer F1 and exact match. The supplementary diagnostic applies a shared, deterministic answer filter to remove explanatory text without consulting the gold answer. It reports token F1 and normalized exact match (EM), complementing the main accuracy score. For a filtered prediction p

Table 10: All-answer coverage at Macro retrieval cutoffs (%). Every normalized gold answer component must occur across the union of the listed candidates. Vector (V) and reranking (R) operate on Macro units; Final Ctx. uses the complete Macro–Micro–chunk input. Bold values are the highest available value within each dataset column.

Dataset	Method/run	V@5	V@10	V@20	V@50	V@100	R@5	R@10	R@20	Final Ctx.
TimeQA	DyG-RAG	29.35	**37.43**	**42.21**	**46.46**	49.87	46.27	56.68	62.07	**76.69**
TimeQA	`IntervalRAG`	**30.04**	33.52	37.05	43.32	**50.21**	**56.95**	**63.57**	**67.97**	74.78
TempReason	DyG-RAG	39.04	**43.60**	**47.51**	**53.99**	**60.29**	76.73	79.30	81.27	97.28
TempReason	`IntervalRAG`	**39.87**	42.28	44.86	50.45	57.51	**77.14**	**80.92**	**81.32**	**97.81**
ComplexTR	DyG-RAG	17.93	19.45	22.19	27.36	30.09	36.17	43.47	44.98	75.99
ComplexTR	`IntervalRAG`	**26.75**	**32.83**	**45.59**	**61.70**	**67.17**	**66.57**	**71.73**	**74.16**	**79.94**

and reference answer a, let T_p and T_a be their normalized token multisets and $o = |T_p \cap_{\text{multi}} T_a|$ their overlap. Token precision, recall, and F1 are

$$P(p,a) = \frac{o}{|T_p|}, \qquad R(p,a) = \frac{o}{|T_a|}, \qquad F_1(p,a) = \frac{2P(p,a)R(p,a)}{P(p,a)+R(p,a)}. \tag{9}$$

We set F1 to zero when either token multiset is empty or there is no overlap. EM requires equality of the entire normalized strings:

$$\text{EM}(p,a) = \mathbb{I}[\text{norm}(p) = \text{norm}(a)]. \tag{10}$$

Dataset scores average the question-level scores and are reported as percentages. F1 credits partial token overlap and penalizes extra tokens; EM requires an exact normalized answer. Neither should be interpreted as the main accuracy measure.

Table 9: Supplementary short-answer diagnostics (%). Each method uses the shared gold-blind answer-filtering protocol.

	TimeQA		TempReason		ComplexTR	
Method	F1	EM	F1	EM	F1	EM
TG-RAG	12.43	7.23	23.02	15.18	21.18	12.46
T-GRAG	21.59	16.65	65.12	52.20	27.08	18.54
DyG-RAG	<u>38.02</u>	**28.17**	<u>79.94</u>	<u>64.98</u>	<u>43.50</u>	<u>23.73</u>
`IntervalRAG`	**43.01**	<u>27.67</u>	**83.10**	**67.82**	**49.77**	**26.75**

In this diagnostic (Table 9), the F1 differences from DyG-RAG are 4.99, 3.16, and 6.27 percentage points across the three datasets; the EM differences are -0.50, 2.84, and 3.02 points. The magnitude and direction of the difference therefore depend on whether evaluation measures answer inclusion, token overlap, or exact answer selection.

All-answer coverage. Let $\mathcal{U}_{q,K}$ be the first K candidates. Each required answer must occur in at least one selected text: $\mathbb{I}[\forall a \in \mathcal{A}_q,\ \exists u \in \mathcal{U}_{q,K} : a \subseteq \text{norm}(x_u)]$. Different components can occur in different events. For example, if two institutions are required and appear in separate candidates, the pair is covered only after both candidates have been included. Final-context coverage applies the same all-answer rule to the actual event and chunk input. These lexical checks do not verify that the matching names describe the correct subject, relation, or time, and can miss unlisted aliases.

I Retrieval-Stage Analysis

We ask whether the retrieved text contains all gold answers. All-answer coverage requires every normalized answer component to occur across the union of the first K candidates; components need not appear in the same event.

Table 10 reports all-answer coverage across Macro retrieval cutoffs. After reranking, IntervalRAG's candidates cover every answer component at R@20 more often than DyG-RAG's on all three datasets: widely on ComplexTR (74.16% versus 44.98%), clearly on TimeQA (67.97% versus 62.07%),

and narrowly on TempReason (81.32% versus 81.27%). Expanding the candidates into the full Macro–Micro–chunk context lifts IntervalRAG's coverage to 97.81% on TempReason and 79.94% on ComplexTR, above DyG-RAG's final-context coverage; on TimeQA the final context reaches 74.78%, just below DyG-RAG's 76.69%. Vector@100 is a diagnostic prefix rather than the full reranking pool.

J Extraction Quality Audit

Corpus-side temporal normalization. During graph construction, `IntervalRAG` normalizes temporal expressions in the corpus into interval representations rather than collapsing every mention into a point timestamp. Concretely:

- **Relative expressions.** Phrases such as "three days later" or "during this period" are anchored to the nearest reliable temporal cue in the same local context.
- **Coarse-grained expressions.** Mentions resolved only to a year, or to a year and month, are conservatively expanded to the full closed interval of that unit, e.g., "in 2012" becomes [2012-01-01, 2012-12-31] and "in March 2012" becomes [2012-03-01, 2012-03-31].
- **Implicit periods.** Open-ended states such as "during tenure" are stored as macro-intervals; missing boundaries are closed using subsequently observed transitions when available.
- **Ambiguous mentions.** If an expression is ambiguous and lacks a reliable textual anchor, the extracted unit keeps a wider interval rather than fabricating a precise date.

Audit protocol. We audit 5,828 extracted events from 275 chunks of **ComplexTR** Tan et al. (2024) using Gemini-2.5-Pro, complemented by small-scale manual spot checks. The audit labels factual support and temporal-boundary quality for individual extracted records; Table 11 reports the results.

Table 11: Extraction-quality audit on ComplexTR. Percentages are computed over the 5,828 extracted events from 275 chunks.

Category	Count	Ratio (%)
Joint correctness (event & time both correct)	4,849	83.20
Time-boundary shift	775	13.30
Format / normalization error	118	2.02
Hallucinated event	86	1.48
Total	**5,828**	**100.00**

The judge labels 83.20% of records jointly correct. Boundary shifts are the largest labeled error category, while hallucinated events account for 1.48%. These results characterize extracted records rather than event recall, and motivate distinguishing uncertain boundaries from unsupported facts.

K Stage-wise Error Propagation

We compare the generator's input context with its response. An input covers a question only when every normalized gold answer component occurs in the combined event and chunk text.

Table 12: Input/output diagnosis of full-response misses. Percentages in the last two columns use the number of misses, not all questions, as denominator.

Dataset	Full-response misses	Input incomplete	Input covers all answers
TimeQA	1,021	648 (63.47%)	373 (36.53%)
TempReason	443	121 (27.31%)	322 (72.69%)
ComplexTR	104	64 (61.54%)	40 (38.46%)

The dominant group differs across datasets (Table 12). TimeQA and ComplexTR have more misses whose inputs do not contain every answer. On TempReason, most misses occur despite complete

lexical answer coverage in the input. The latter group is not necessarily pure reasoning failure: the matching strings can occur with the wrong subject, relation, or date, or amid distracting evidence. Likewise, lexical absence does not prove unanswerability because aliases and paraphrases can evade the matcher.

Ordered trace diagnostics. We further localize each full-response miss to the first stage at which the required answer components are lost: Macro all-answer coverage at Vector@100, then reranking, then the actual final context, then the generated output. Because each miss is attributed to its first failed stage, the output bin is smaller than the input-coverage count of Table 12: it counts only misses that remain fully covered at every retrieval stage, whereas an answer that is absent from the retrieved candidates yet recovered during context assembly is assigned to the earlier stage. The resulting counts are shown in Table 13.

Table 13: Ordered diagnostic bins among full-response misses. Labels denote the first failed *check*, not a causal attribution.

Dataset	V@100 miss	R@K miss	Context miss	Output miss
TimeQA	747 (73.16%)	103 (10.09%)	18 (1.76%)	153 (14.99%)
TempReason	314 (70.88%)	42 (9.48%)	8 (1.81%)	79 (17.83%)
ComplexTR	77 (74.04%)	3 (2.88%)	1 (0.96%)	23 (22.12%)

The first failed stage is dominated by retrieval on every dataset: Vector@100 accounts for 73.16%, 70.88%, and 74.04% of misses on TimeQA, TempReason, and ComplexTR, meaning that in most failures the answer components are already absent from the candidate pool. Misses introduced later are smaller and differently distributed: reranking contributes between 2.88% and 10.09%, context assembly stays below 2% throughout, and generation-stage misses are largest on ComplexTR (22.12%) and smallest on TimeQA (14.99%). Full-response misses therefore reflect both evidence retrieval and answer selection, with the balance shifting toward generation on the multi-hop ComplexTR questions.

L Efficiency Comparison

We report accuracy (**Acc.**, %) alongside offline indexing time (**IT**, in 10^3 seconds) and average online query time (**QT**, in seconds); Table 14 lists the results. Indexing uses an $8\times$RTX 3090 server; query time includes generation and is not pure retrieval latency.

Table 14: Efficiency comparison. **Acc.** is accuracy (%); **IT** is offline indexing time in 10^3 s; **QT** is average online query time in seconds.

Method	**TimeQA** Chen et al. (2021)			**TempReason** Tan et al. (2023)			**ComplexTR** Tan et al. (2024)		
	Acc. ↑	**IT** ↓	**QT** ↓	**Acc.** ↑	**IT** ↓	**QT** ↓	**Acc.** ↑	**IT** ↓	**QT** ↓
GraphRAG-L Edge et al. (2024)	40.26	56.30	81.71	56.11	44.61	55.21	43.16	3.75	20.77
LightRAG-H Guo et al. (2024)	36.36	8.05	26.31	50.01	11.12	23.73	42.68	1.34	12.38
HippoRAG Jimenez Gutierrez et al. (2024)	39.99	6.07	3.54	69.80	15.72	15.72	44.68	1.23	18.73
E^2GraphRAG Zhao et al. (2025)	40.48	36.58	140.20	61.29	63.63	79.39	38.29	5.39	72.45
DyG-RAG Sun et al. (2025)	58.78	13.96	50.49	84.75	19.62	53.95	55.62	1.54	34.60
`IntervalRAG`	60.93	18.70	20.88	91.79	24.50	21.14	68.39	23.10	21.59

`IntervalRAG` has lower query time than DyG-RAG on all three datasets, with higher offline indexing time. The preprocessing overhead is largest on ComplexTR (23.10 versus 1.54 thousand seconds). Indexing can be amortized when a corpus is reused, but the amortization point depends on query volume and serving conditions.

Incremental updates. In principle, new documents can be incorporated by applying the same extraction and indexing pipeline locally, although we leave a full incremental-indexing evaluation to future work. For each new chunk, `IntervalRAG` can extract and normalize macro-interval and micro-event units, insert them into the corresponding temporal buckets and macro containers, and update the affected vector and graph index entries.

M QUALITATIVE CASE STUDIES

The following examples illustrate one case from each benchmark. To complement the quantitative results in Section 4, we present three end-to-end qualitative case studies (one per benchmark) that contrast the reasoning trace of DyG-RAG with that of `IntervalRAG`. The cases are chosen to highlight three recurring failure modes of point-based temporal RAG, namely *gap-filling hallucination*, *missed implicit state*, and *noise-driven dropout*, and to show how interval-based retrieval addresses each of them. In every case, red marks incorrect answers or misleading evidence, while green marks correct answers and valid temporal supporting events.

M.1 TEMPREASON: GAP-FILLING HALLUCINATION

Table 15: Qualitative comparison between DyG-RAG and `IntervalRAG` on a TempReason Tan et al. (2023) case in which the baseline fills a missing tenure window with the next-known incumbent.

Dataset: TempReason Tan et al. (2023)

Question: Who was the head of Romania in Jan, 1997?

Ground truth: Victor Ciorbea

Source text: Victor Ciorbea is the head of the government of Romania from Dec, 1996 to Apr, 1998. Nicolae Văcăroiu became the head . . . from November 1992 to December 1996.

DyG-RAG.

Event 30 [1992-11]: Nicolae Văcăroiu is the head of the government of Romania from Nov, 1992 to Dec, 1996.

Event 31 [1998-04]: Radu Vasile is the head of the government of Romania from Apr, 1998 to Dec, 1999.

Temporal analysis. There is a gap between December 1996 and April 1998. Assuming continuity and given the absence of other named heads, Radu Vasile must have been the head during this gap.

Final answer. Radu Vasile was the head of the government of Romania in January 1997.

`IntervalRAG` (ours).

Macro Interval 1. Victor Ciorbea was the head of the government of Romania from December 1996 to April 1998. Time period: [1996-12-01, 1998-04-30].

Local timeline (micro-events): **[E1]** [1996-12-01, 1998-04-01] Victor Ciorbea became the head of the government of Romania. → *Allen relation [*MEETS*]* → **[E2]** [1998-04-01] Radu Vasile became the head of the government.

Macro Interval 3. Nicolae Văcăroiu was the head from November 1992 to April 1998, followed by Victor Ciorbea. Time period: [1992-11-01, 1998-04-30].

Local timeline (micro-events): **[E4]** [1992-11-01, 1996-12-01] Nicolae Văcăroiu became the head. → *Allen relation [*MEETS*]* → **[E5]** [1996-12-01, 1998-04-01] Victor Ciorbea became the head of the government of Romania.

Timeline analysis. Multiple macro-intervals confirm that Victor Ciorbea's tenure spanned December 1996 to April 1998, which directly covers the query date of January 1997.

Final answer. Victor Ciorbea was the head of Romania in Jan, 1997.

Analysis of Table 15. In this case, the DyG-RAG evidence is centered on sparse point anchors and does not preserve the mid-interval tenure needed for January 1997. This opens a gap between December 1996 and April 1998, making the retrieved evidence weak support for the queried time window and leading the generator to select the next-known incumbent. `IntervalRAG` resolves the query via *interval containment*: macro-anchoring directly surfaces a macro-interval whose span covers January 1997, and logic-guided traversal recovers micro-events that explicitly connect successive state transitions. The Allen MEETS relation records the boundary alignment between successive heads, and multiple corroborating macro-intervals provide independent validation of the same tenure window.

M.2 ComplexTR: Implicit-State Recovery

Table 16: Qualitative comparison between DyG-RAG and `IntervalRAG` on a ComplexTR Tan et al. (2024) case in which the baseline fails to recover an implicit postdoctoral state.

Dataset: ComplexTR Tan et al. (2024)

Question: Where was Jiankui He educated in July 2011?

Ground truth: Stanford University

Source text: He entered Rice University in 2007 and received his Ph.D. degree from … in 2010. After his Ph.D., Deem arranged for He to work on … at Stanford University. He returned to China in 2012…

DyG-RAG.

Event 6 [1984]: He Jiankui was born in Xinhua County, Loudi, Hunan in 1984.

Event 11 [2010]: He Jiankui earned his Ph.D. from Rice University in 2010.

Event 14 [2012]: He Jiankui founded Direct Genomics in 2012.

Event 15 [2012]: He Jiankui joined the Department of Biology of SUSTech as an associate professor in 2012.

Temporal analysis. He Jiankui moved to Rice University in 2007 and completed his Ph.D. in 2010. Between his undergraduate studies and his Ph.D., there is no mention of further education or employment that would indicate he was elsewhere.

Final answer. He was preparing for his Ph.D. at Rice University.

`IntervalRAG` (ours).

Macro Interval 1. As a postdoctoral researcher at Stanford University, He Jiankui worked… He returned to China in 2012… Time period: [2010-01-01, 2018-01-01].

Local timeline (micro-events): **[E1]** [2012-01-01] He Jiankui returned to China in 2012 under the Thousand Talents Program. → *Allen relation [*EQUALS*]* → **[E2]**… **[E3]**… **[E4]**…

Macro Interval 2. He Jiankui earned his Ph.D. in physics and astronomy from Rice University, focusing on protein evolution including CRISPR, under the supervision of Michael W. Deem. Time period: [2006-01-01, 2010-01-01].

Local timeline (micro-events): **[E5]** [2010-01-01] He Jiankui earned his Ph.D. from Rice University in 2010. → *Allen relation [*STARTS*]* → **[E6]** [2010-01-01, 2012-01-01] He Jiankui worked as a postdoctoral researcher at Stanford University from 2010 to 2012.

Timeline analysis. He Jiankui completed his Ph.D. at Rice University in 2010 and started his postdoctoral position at Stanford in 2010, continuing until 2012. Therefore, during the period [2011-07-01, 2011-07-31] he was a postdoctoral researcher at Stanford University.

Final answer. Stanford University

Analysis of Table 16. In this case, the DyG-RAG evidence is centered on explicit point-like events and does not preserve the implied state between 2010 and 2012. The retrieved records therefore provide weaker support for the July 2011 query window, leading the answer toward the closest preceding institution (Rice University). `IntervalRAG` uses macro-anchoring to surface the postdoctoral interval at Stanford, whose source text gives its span as 2010 to 2012, so the July 2011 window falls inside it; the Allen STARTS relation then records the adjacency between the Ph.D. event and the start of that interval rather than establishing the state between them.

M.3 TIMEQA: NOISE-DRIVEN DROPOUT

Table 17: Qualitative comparison between DyG-RAG and `IntervalRAG` on a TimeQA Chen et al. (2021) case in which the baseline returns date-matching but topically irrelevant evidence.

Dataset: TimeQA Chen et al. (2021)
Question: Which team did the player Steve Sidwell belong to between May 2010 and Sep 2010?
Ground truth: Aston Villa
Source text: On 10 July 2008, Sidwell signed a three-year deal with Aston Villa . . . Sidwell made 33 appearances . . . during the 2009/10 season. Sidwell missed the start of the 2010/11 season with an injury . . . Sidwell signed for Fulham on 7 January 2011.

DyG-RAG.
Event 44 [2010-05-12]: Linwood was one of seven players . . . (irrelevant player).
Event 49 [2010-07-29]: Andy Schleck announced departure . . . (irrelevant athlete).
Event 66 [2011-01-07]: Sidwell signed for Fulham on 7 January 2011.
Analysis. There is no mention of Sidwell's team between May and July 2010. Since he only signed for Fulham in 2011, it implies he was not playing for any club during the period.
Final answer. Steve Sidwell did not belong to any team.

`IntervalRAG` (ours).
Macro Interval 4. [2009-07-01, 2010-06-30] Sidwell played for Aston Villa during the 2009/10 season.
Macro Interval 9. [2010-07-01, 2011-06-30] Sidwell was with Aston Villa for the 2010/11 season (missing the start due to injury) before joining Fulham in Jan 2011.
Local timeline: **[E12]** [2009-08 / 2010-05] Sidwell made 33 appearances for Villa. $\rightarrow$ *Allen relation [*BEFORE*]* $\rightarrow$ **[E27]** [2010-08-01] Sidwell missed the start of the season at Villa due to injury.
Timeline analysis. Sidwell's tenure at Aston Villa spanned from his 2008 contract until his 2011 transfer. The window from May 2010 to September 2010 is the transition between the 2009/10 and 2010/11 seasons, and the injury at the "start of the 2010/11 season" confirms that he was still a Villa player.
Final answer. Aston Villa

Analysis of Table 17. In this case, the DyG-RAG evidence is centered on point-like events around the query dates and includes records that are date-matched but topically irrelevant (e.g., Linwood and Andy Schleck). The retrieved records therefore provide weaker support for Sidwell's team membership during the queried window, leading to an unsupported teamless answer. `IntervalRAG` identifies long-duration tenures as macro-interval units, and the retrieved source evidence—the 2008 contract, the 2011 transfer, and the injury at the start of the 2010/11 season—grounds Sidwell's continued membership across the off-season. The Allen relations organize the retrieved events into a local chronology rather than establishing that the state persisted through the gap.

N PROMPT TEMPLATES

The indexing stage applies two complementary extraction passes to each source chunk. The macro pass groups related facts into source-grounded phases, while the micro pass records atomic, time-localized facts that preserve details needed for retrieval. Both passes use the same source text and return structured JSON. The answer-generation template then receives the question together with the retrieved events and source chunks; it specifies how to collect candidates, compare their temporal support, and format the final answer. The templates below show the input/output contracts and the decision rules that affect reproducibility.

N.1 MACRO-INTERVAL EXTRACTION PROMPT

Prompt A Macro-Interval Extraction

ROLE

You are an expert *Timeline Analyst*. Your task is to extract Macro-Interval Units from the input text.

DEFINITION: MACRO-INTERVAL UNIT

A Macro-Interval Unit is a long-duration state, phase, role, affiliation, or temporal context that can contain multiple smaller micro-events.

- **Scope.** Months or years, such as early life, education, career tenure, project lifecycle, or later years.
- **Synthesis.** Read the detailed input and synthesize a self-contained summary for each interval.

INTERVAL COVERAGE

The collection of extracted macro-intervals should cover the full chronological span of the input text.

- **Start point.** The first interval starts at the earliest relevant date mentioned.
- **End point.** The last interval ends at the latest relevant date mentioned.
- **Continuity.** Extract transition phases when necessary to preserve the continuity of long-duration states.

TIME INFERENCE RULES

- Use strict ISO-8601 dates in the format `YYYY-MM-DD`.
- If only the year is known, normalize the start to `YYYY-01-01` and the end to `YYYY-12-31`; if only the year and month are known, span that month from `YYYY-MM-01` to its last day.
- Avoid open ends whenever the text provides enough evidence; infer a specific end date from the start of the next phase if necessary.
- The start and end dates must define the boundaries of the entire phase.

OUTPUT FORMAT

Return a valid JSON object with an `"events"` key containing a list of objects, each with the following fields.

```
{
  "events": [
    {
      "summary":    "Subject + State/Role/Location + Organization + Key Evidence +
     Ending Context",
      "start_time": "YYYY-MM-DD",
      "end_time":   "YYYY-MM-DD"
    }
  ]
}
```

INPUT

`{input_text}`

OUTPUT

JSON only.

N.2 MICRO-EVENT EXTRACTION PROMPT

Prompt B Micro-Event Extraction

ROLE

You are a meticulous *Event Extraction Engine*. Your task is to extract all *Micro Events* (specific actions, interactions, and milestones) from the text.

DEFINITION: MICRO EVENT

A Micro Event is a specific, atomic action, interaction, transition, or milestone. Although each event is atomic, it is represented by a time interval rather than a single timestamp: an instantaneous event collapses to a point, while a coarser or duration-bearing mention keeps its supported span.

- **Goal.** High recall: extract every action, change, or milestone.
- **Atomicity.** If a sentence contains multiple actions, extract them as separate events; do not merge them.
- **Content.** The `content` field must be detailed and self-contained.

CRITICAL RULES

1. **Subject resolution.** Replace pronouns with the specific entity name.
2. **Time inference.** Use ISO-8601 dates in the format `YYYY-MM-DD`. When a mention resolves only to a coarser calendar unit, cover that whole unit: a year-only mention sets `start_time` to `YYYY-01-01` and `end_time` to `YYYY-12-31`, and a month-only mention spans the first to the last day of that month. A full date with no stated duration is a point (`end_time` equal to `start_time`).
3. For implicit or underspecified times, infer the event time from the local temporal context: paragraph-level dates, preceding time anchors, event order, and stated time spans.
4. Resolve relative or anaphoric expressions such as "three days later", "that year", "the next day", "before", or "after" using the nearest valid temporal anchor in the text.
5. Do not invent dates from outside knowledge; every inferred time must be grounded in the input text.

OUTPUT FORMAT

Return a valid JSON object with an `"events"` key containing a list of objects, each with the following fields.

```
{
  "events": [
    {
      "content":    "Subject + Action + Object + Context",
      "start_time": "YYYY-MM-DD",
      "end_time":   "YYYY-MM-DD"
    }
  ]
}
```

For instantaneous events, set `end_time` equal to `start_time`.

INPUT

`{input_text}`

OUTPUT

JSON only.

Prompt C Answer-Generation Prompt

```
You are a temporal QA finalization engine. Answer the user query using only the
supplied Events and Text Chunks, producing reasoning followed by a final answer.
The Final Answer line is a copy step: it must exactly reproduce the Answer set
line written in the reasoning, so no new decision is made after the Answer set.

Evidence priority:
1. Use only the supplied Events and Text Chunks; do not use outside knowledge.
2. Treat Text Chunks as the highest-priority evidence, since Events may carry
   approximate normalized dates; if the two conflict, trust the Text Chunks.
3. Do not invent names, aliases, dates, or roles: every output entity must appear
   in the supplied evidence.

Task setup:
1. Identify the subject of the query and the requested relation type (employer,
   educated-at, spouse, political party, position/title, team, residence, award,
   or other).
2. Resolve the target date or interval; for relative questions, compute it from
   the trigger event, and keep it approximate when the trigger date is approximate.
3. Extract all candidates for the same subject and relation from both Events and
   Text Chunks before deciding the final answer.

Temporal decision labels:
- EXACT-IN: the support clearly contains the target date or interval.
- FUZZY-IN: the support is year-only, approximate, sequential, or normalized.
- AMBIGUOUS-IN: the candidate is same-subject and same-relation, and the context
  does not rule it out for the target.
- EXACT-OUT: the support proves the candidate ended before or started after the target.
- WRONG: wrong subject or relation, a trigger entity, or an unsupported mention.

Inclusion policy:
1. Include EXACT-IN, FUZZY-IN, and AMBIGUOUS-IN candidates, and exclude EXACT-OUT
   or WRONG candidates.
2. Include any candidate whose interval mathematically contains the target; do
   not reject an interval such as "from 1926 to 1934" for the target 1931.
3. Treat year-only evidence as covering the whole calendar year, and treat
   YYYY-01-01 and YYYY-12-31 as fuzzy normalized boundaries unless the text states
   that exact month and day.
4. Keep one Candidate line per real entity or title, and preserve aliases and
   translations separated by " | ".

Reasoning format:
Reasoning:
- Target: the subject, the requested relation, and the target date or interval.
- Candidate ledger: one entry per candidate, each with a brief Support quote or
  paraphrase, a Time comparison against the target, Relation match (YES/NO), Time
  status, and Final include (YES/NO).
- Final consistency check: confirm that every Final include YES candidate appears
  in the Answer set and every NO candidate does not.
- Answer set: <one line of all included names and aliases separated by " | ", or
    Unknown>

Final Answer:
<copy the Answer set line exactly>

Final Answer line rules:
- Write only entity or title names, or Unknown; list complete names first, then
  any aliases, translations, or title variants found in the evidence.
- Do not write a sentence, and do not include dates, roles, evidence, or
  uncertainty text.

INPUT DATA

User Query:
{question}

Events:
{events_data}

Text Chunks:
{chunks_data}

RESPONSE
```

O LIMITATIONS AND FUTURE WORK

We discuss the principal limitations of `IntervalRAG` and outline concrete future-work directions that follow from them.

Dependence on automatic temporal extraction. `IntervalRAG` relies on an offline LLM-based extractor to populate the hierarchical temporal graph. As shown in Appendix J, the dominant residual errors are temporal-boundary shifts rather than event hallucinations, but noisy or missing interval boundaries can still affect graph construction and early-stage evidence recall, especially when source documents contain ambiguous, sparse, or contradictory temporal expressions. *Future work:* explore (i) extractor confidence calibration so that uncertain intervals are widened automatically, and (ii) joint extraction and retrieval training that allows downstream signals to refine upstream interval boundaries.

English-centric Allen-relation parsing. The current Allen-relation predicates and the query-recovery procedure (Appendix G) are derived from English temporal cues such as "before", "after", "during", and "two years before". Extending the framework to multilingual or domain-specific time expressions (e.g., legal, biomedical, or financial calendars) is non-trivial and is left to future work.

Indexing overhead at scale. Although the coarse-to-fine retrieval procedure is efficient at query time (Appendix L), large macro-interval graphs may incur non-trivial offline indexing overhead. Incremental indexing appears feasible by applying extraction and indexing locally to new documents, but full-corpus rebuilds for very large or rapidly evolving corpora may remain expensive. *Future work:* investigate hierarchical pruning of redundant macro-intervals and approximate, partition-based graph construction.

Scope of evaluation. Our experiments focus on three temporal QA benchmarks (TimeQA, TempReason, ComplexTR). While these benchmarks collectively cover implicit reasoning, state localization, and multi-hop reasoning, they do not exhaust the full space of time-sensitive generation tasks. *Future work:* extend interval-aware retrieval to long-form temporal summarization, temporal fact verification, and event-forecasting style tasks, where evidence chronology directly drives generation.

Pipeline-level retrieval bottleneck. The error-propagation analysis (Appendix K) shows that the dominant remaining error mode is early evidence recall: the answer-bearing chunk is sometimes not in the initial candidate set returned by vector retrieval. *Future work:* combine `IntervalRAG` with stronger temporally-conditioned retrievers and rerankers, and explore query-time graph expansion to recover gold evidence missed by the initial macro-anchoring step.

P DECLARATION OF LLM USAGE

We describe the roles that LLMs play in the method, evaluation, and manuscript preparation.

LLMs as part of the proposed method. LLMs are an integral component of the `IntervalRAG` pipeline. Specifically, an LLM is used (i) at indexing time to extract macro-interval and micro-event units from raw chunks (Prompts A and B in Appendix N) and (ii) at inference time as the final answer generator conditioned on the structured evidence assembled by the retrieval and traversal modules (Prompt C). The backbone model, exact prompts, and decoding configuration are reported in Appendices C and N, so that the LLM-dependent parts of the method can be reproduced.

LLMs for evaluation. For the extraction-quality audit in Appendix J, we use Gemini-2.5-Pro as an LLM-as-a-judge over a sample of 5,828 extracted events, complemented by small-scale manual spot checks. End-task accuracy and recall (Section 4) are computed deterministically with the rule-based string matching described in Appendix H and do not involve any LLM-based scoring.

LLMs for writing assistance. Generative AI tools assisted with language editing, manuscript organization, and consistency checks. The authors are responsible for the final manuscript, scientific claims, and reported results.