

EventSocraticKG: Event-Centric Knowledge Graph Construction for Narrative Retrieval-Augmented Generation

William Rosengren

KTH Royal Institute of Technology

wrose@kth.se

Abstract

Graph-based retrieval over book-length narrative text faces a choice at construction time. Merging every mention of a character into one node erases the temporal context that separates an early portrayal from a later one; keeping mentions apart scatters a character across disconnected per-chunk nodes and leaves the consolidation to query time. We present **EventSocraticKG** (eSoKG), which (i) makes events first-class nodes in a bipartite event–entity graph, (ii) extracts them from 5W1H question–answer pairs written for each chunk instead of from the chunk text, and (iii) merges mentions across chunks by their full description, with embeddings proposing candidates and an LLM confirming them. At query time eSoKG serves graph structure only, with no source chunks and no LLM call. On 332 long-form questions over five books (four pre-2000 novels and one post-cutoff novel), scored by an LLM judge and re-scored by two independent judges, eSoKG lies within half a point of an unmerged entity–event comparator and of a full-budget entity-centric method that spends roughly twice the per-query tokens, under all three judges. The post-cutoff book separates retrieval from recall: an LLM without retrieval collapses there while every retrieval method holds, but eSoKG trails both graph comparators, so part of its pooled standing reflects books the models already know. Its category profile, strongest on the consistency categories and weakest on Narrative and Thematic, is largely unchanged on that book. The price is a one-time build two to three times the comparators’. What it buys is a chunk-free query path in the low per-query cost regime and a merged graph on which character-chain, shared-event, and causal-chain queries are short graph walks that need no language model.

1 Introduction

A novel is a sequence of events whose meaning depends on everything that came before. Answering questions about the first Harry Potter book, such as

“which characters appear in scenes with both Harry and Snape during his first year?” or “what earlier events lead the trio to the trapdoor?”, requires tracing chains of events across the whole book rather than looking up a single passage. A knowledge graph (KG) that records events, their participants, and the causal–temporal links between them does this structural work once at construction time and exposes it cheaply at every query. Graph-based retrieval-augmented generation (RAG) builds on this, and so do agents that navigate the graph directly (Ghandi et al., 2026; Ni et al., 2025). In both cases the answer can only be as good as the graph, which puts the weight on how the graph is built.

Published methods for building a KG over book-length text share the same recipe: split the source into chunks, extract a subgraph per chunk, and combine the subgraphs. The combination step is where things get hard. The same character is named differently across chunks, and reconciling those mentions raises a dilemma. *Merging* every mention into one node, as entity-centric methods do by name, can average away a character whose state drifts across the story. *Not merging*, the choice made by the closest prior work, E²RAG (Zhang et al., 2026), to preserve that temporal context, leaves a character as many disconnected nodes, so a query about “every event this character took part in” must first gather all the fragments back together, doing at query time the consolidation that construction skipped.

We take the opposite position from E²RAG and ask whether the dilemma is false: whether a character’s *per-event participations* can carry the temporal context that keeping mentions separate protects, so that a merged graph answers narrative questions as well as an unmerged one, but more cheaply per query. Our method, **EventSocraticKG** (eSoKG), makes three design choices that no prior system combines:

- an **event-centric bipartite schema** with no entity–entity edges;
- **5W1H question-mediated extraction**, adapted from SocraticKG (Choi et al., 2026), which turns each chunk into question–answer pairs before any structure is extracted, to surface participant roles and causal links that direct extraction misses;
- **cross-chunk merging by full description**, using embeddings to propose merge candidates and an LLM to confirm them, rather than string-matching names.

At query time eSoKG retrieves a *chunk-free* context: distilled entity, event, and relation descriptions only, with no raw source text and no LLM call. We evaluate the resulting graph inside a RAG pipeline and ask two questions:

RQ1 (quality) Does a chunk-free structural context answer narrative questions as accurately as graph-based methods whose context includes the raw source chunks?

RQ2 (cost) How does the approach compare in cost, both the per-query cost of serving a question and the one-time cost of building the graph?

Separately from this QA evaluation, we argue for a benefit it does not measure: because events are first-class nodes and merging consolidates each entity into one, the graph answers structural queries (a character’s event chain, the events two characters share, the causal chain into an event) as short graph walks needing no language model, which the entity-centric and unmerged comparators cannot serve directly.

Against four comparators on the ChronoQA benchmark plus one post-cutoff novel, eSoKG lies within half a point of the best method on the pooled question set under all three judges, at a low per-query cost, and it is the only method that spends no LLM tokens at retrieval time. On the single contamination-free book it trails both graph comparators, and its one-time build cost is two to three times the comparators’.

2 Background and Related Work

From triples to events. The triple (s, p, o) fits binary facts but forces a narrative event, with several participants in distinct roles, anchored in time and linked to causes and effects, into either compound nodes specific to one passage (coverage, at

the cost of reuse) or a set of pairwise edges that dissolves the event’s unity (coherence, at the cost of the event) (Choi et al., 2026). Hyperedges resolve this by connecting any number of entities, and their bipartite reformulation, in which each hyperedge becomes a node joined by ordinary edges to its participants, keeps n -ary relations while staying compatible with standard graph tooling (Luo et al., 2025; Wen et al., 2016). Event-centric KGs (Bai et al., 2025) take this form: events are nodes, and edges record participation and event–event causality.

LLM-driven construction. Prompting an instruction-tuned LLM with a passage and a schema yields entities, relations, and events directly (Wei et al., 2024; Edge et al., 2025; Guo et al., 2025; Bai et al., 2025). The open question is whether the graph faithfully captures the source. No gold reference graph exists for a book-length narrative, so the question has to be answered *extrinsically*, by placing the graph in a RAG pipeline and scoring downstream answers.

QA-mediated extraction and deduplication. A separate strand writes content out as natural-language question–answer pairs before extraction; SocraticKG (Choi et al., 2026) argues the 5W1H structure of that step surfaces motivations and roles that single-pass extraction skips. Cross-chunk deduplication (Christen, 2012) must answer two questions, *when* two mentions co-refer and *whether* they should be merged, because merging a referent with distinguishable states erases what separates them.

Positioning. Table 1 places eSoKG against its three closest comparators. It inherits E²RAG’s bipartite event-centric schema and target use case but takes the opposite stance on merging; it adapts SocraticKG’s 5W1H step but places it inside an event schema rather than a flat triple graph; and unlike LightRAG it drops entity–entity edges and name-based merging entirely. EventRAG (Yang et al., 2025b) also consolidates events by similarity but targets multi-hop QA over knowledge-heavy collections with dated edges, not the unlabelled state-drift of a single narrative.

Axis	LightRAG	E ² RAG	eSoKG (ours)
Focus	general	narrative	narrative
Schema	entity-rel.	bipartite ev.	bipartite ev.
Extract	1-2 calls/chunk	2 calls/chunk	5W1H QA → ent. → ev.
Dedup	by name	none	by description (emb.+LLM)
Context	graph+chunks	graph+chunks	chunk-free

Table 1: eSoKG against its closest comparators. SocraticKG (not shown) contributes the 5W1H QA step but outputs a flat triple graph evaluated intrinsically.

3 EventSocraticKG

3.1 Graph

For a book b , the eSoKG graph is

$$G_{\text{eSoKG}}^{(b)} = (V_{\text{ent}} \cup V_{\text{evt}}, E_{\text{evt}} \cup B),$$

with entity nodes V_{ent} , event nodes V_{evt} , typed directed event-event relations $E_{\text{evt}} \subseteq V_{\text{evt}} \times V_{\text{evt}}$ (causal and temporal), and a bipartite participation relation $B \subseteq V_{\text{ent}} \times V_{\text{evt}}$. There are *no* entity-entity edges: entities relate only through shared events, which makes the event chains in E_{evt} the load-bearing structure for narrative reasoning. Nodes and edges carry types from closed vocabularies (10 entity types; 12 event types such as Movement, Decision, Conflict; 9 relation types such as causes, enables, follows), passed to the LLM as guidance; node types have an Other fallback.

3.2 Indexing

Indexing runs in two phases: local per-chunk extraction, then global reconciliation. The source is split with a token-based sliding window (2000-token chunks, 400 overlap), and each chunk is keyed by its content hash and processed independently.

Phase 1: QA-mediated per-chunk extraction.

Each chunk goes through three sequential LLM calls. (1) *QA generation*: the chunk is turned into self-contained, pronoun-resolved question-answer pairs; the prompt uses the 5W1H questions to drive coverage of the chunk, not as a quota. (2) *Entity extraction*: a second call reads *only the QA pairs* (never the raw chunk) and returns entity records (name, type, aliases, a 1-3 sentence description) tagged with ephemeral identifiers. (3) *Event extraction*: a third call receives the QA pairs and a slim view of the entities and returns events (each

with participants referenced by the ephemeral tags) together with intra-chunk event-event relations. Splitting entity and event extraction costs one extra round-trip but lets the event extractor reference a stable entity list, producing cleaner participant lists. Bipartite edges are then added from each event’s explicit participant list, with a case-insensitive substring fallback scanning the event description for known entity names (the same rule E²RAG uses as its only mechanism). Mentions receive chunk-scoped stable identifiers, so two mentions of one name in two chunks stay distinct until Phase 2.

Phase 2: description-based deduplication.

After Phase 1 a character is typically spread over many nodes. Phase 2 merges duplicates using two signals. Embeddings are cheap over all pairs but capture only surface form; an LLM can judge true co-reference but is too costly per pair. eSoKG therefore lets *embeddings propose and the LLM confirm*: a merge needs both a candidate pair from the embeddings and a yes from the LLM. Each node is embedded from its name, type, description, and aliases (entities) or participants (events). At a similarity threshold τ , connected components of the thresholded pairwise-similarity graph form candidate clusters; multi-member clusters go to an LLM told the candidates came from vector similarity, that none is privileged, and that returning zero merges is valid. The threshold is swept high-to-low so safe merges commit and enrich descriptions before looser candidates are examined, and the whole sweep is run *twice*, re-embedding merged clusters from their consolidated descriptions to catch duplicates whose original vectors were too narrow. Event thresholds are kept higher than entity thresholds, and the event prompt is told that recurring events with overlapping participants are still distinct, because a false event merge permanently destroys a distinction whereas a miss only leaves a duplicate. Commits are also *asymmetric*: an absorbed entity’s chunk-scoped vector is kept, with a `merged_with` pointer, so it still works as a retrieval entry point, while an absorbed event’s vector is deleted as noise. All stored identifiers (event participants, E_{evt} , and B) are finally rewritten to canonical ids.

3.3 Chunk-free retrieval

Retrieval makes *no* LLM call. The question is embedded once and reused for two parallel seeding paths. *Path A (event-first)* queries the event store

Algorithm 1 eSoKG indexing**Require:** Book b

```

1: Phase 1: per-chunk extraction
2: for all  $c \in \text{CHUNK}(b)$  do
3:    $qa \leftarrow \text{QALLM}(c)$   $\triangleright$  5W1H QA pairs
4:    $V_{\text{ent}}^c \leftarrow \text{ENTITYLLM}(qa)$ 
5:    $V_{\text{evt}}^c, E_{\text{evt}}^c \leftarrow \text{EVENTLLM}(qa, V_{\text{ent}}^c)$ 
6:    $B \leftarrow B \cup \text{LINK}(V_{\text{ent}}^c, V_{\text{evt}}^c)$ 
7:   accumulate  $V_{\text{ent}}, V_{\text{evt}}, E_{\text{evt}}$ 
8: end for
9: Phase 2: global deduplication
10:  $V_{\text{ent}}^* \leftarrow \text{DEDUP}(V_{\text{ent}})$ ;  $V_{\text{evt}}^* \leftarrow \text{DEDUP}(V_{\text{evt}})$ 
11: rewrite participants,  $E_{\text{evt}}, B$  to canonical ids
12: return  $(V_{\text{ent}}^* \cup V_{\text{evt}}^*, E_{\text{evt}} \cup B)$ 

```

Algorithm 2 DEDUP(V): embeddings propose, LLM confirms**Require:** nodes V ; thresholds $\tau_1 > \dots > \tau_k$

```

1:  $M \leftarrow$  row-normalised embeddings of  $V$ ;  $S \leftarrow MM^\top$ 
2: for pass  $\in \{1, 2\}$  do
3:   for  $\tau \in \{\tau_1, \dots, \tau_k\}$  do
4:      $G_\tau \leftarrow \{(i, j) : S_{ij} \geq \tau\}$  over current canonicals
5:     for all multi-member component  $K$  of  $G_\tau$  do
6:       for all  $K' \in \text{LLMVERIFY}(K)$  do
7:          $\text{MERGE}(K')$   $\triangleright$  survivor absorbs the rest
8:       end for
9:     end for
10:   end for
11:   if pass = 1 then re-embed merged clusters; recompute  $S$ 
12:   end if
13: end for
14: return canonical  $V^*$ 

```

directly for the top- k events ($k=10$). *Path B* (entity-first) queries the entity store, canonicalises each hit via its merged_with pointer, walks B to gather every event the entity participates in, and reranks that (often large) set by cosine to the query, keeping the top 10. The two seed sets are unioned A-first. The seeds are then expanded by an undirected breadth-first walk over E_{evt} to depth two, since a cause of a retrieved effect is as relevant as an effect of it, though edge direction is preserved in the rendered context. The expanded set is truncated to a 15k-token budget by keeping the largest breadth-first prefix that fits, preserving narrative locality. The context is assembled as three sections: an entities table, an events table, and an event-relationships list emitted in connected-component order so chains stay together. No raw source text is included.

3.4 Algorithmic summary

Algorithms 1–3 state indexing, deduplication, and retrieval; LINK is the bipartite construction of §3.2. Default thresholds and batch sizes are in Appendix A.

Algorithm 3 eSoKG retrieval (no LLM call)**Require:** question q , graph G , budget β

```

1:  $\mathbf{v}_q \leftarrow \text{EMBED}(q)$ 
2:  $S_{\text{ent}}^A \leftarrow \text{TOPK}(V_{\text{ent}}^*, \mathbf{v}_q)$   $\triangleright$  Path A
3:  $S_{\text{ent}} \leftarrow \text{CANON}(\text{TOPK}(V_{\text{ent}}^*, \mathbf{v}_q))$ 
4:  $S_{\text{evt}}^B \leftarrow \text{TOPK}(\text{EVENTSFOR}(S_{\text{ent}}, B), \mathbf{v}_q)$   $\triangleright$  Path B
5:  $E \leftarrow \text{EXPAND}(S_{\text{ent}}^A \cup S_{\text{evt}}^B, E_{\text{evt}})$   $\triangleright$  2-hop, undirected
6:  $E^* \leftarrow \text{TRUNCATE TO BUDGET}(E, \beta)$ 
7: return  $\text{ASSEMBLE}(E^*, B, E_{\text{evt}})$ 

```

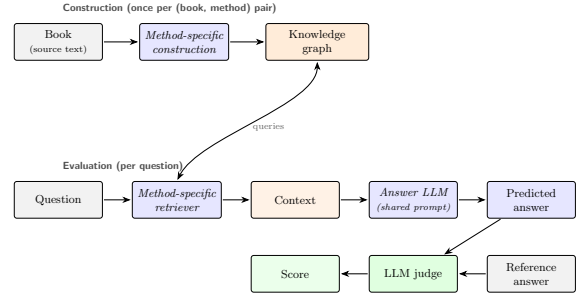


Figure 1: The evaluation. A graph is built once per (book, method) pair and queried per question. The answer LLM, its prompt, and the judge are shared across methods, so the only variable is the graph and its retriever.

4 Experimental Setup

Corpus. We evaluate on 332 long-form questions over five books (Table 2). Four are novels from ChronoQA (Zhang et al., 2026), the narrative-QA benchmark released with E²RAG, chosen to span a range of lengths and question counts; each question is tagged with one of eight reasoning categories (e.g. Causal Consistency, Character Consistency). The fifth, *The Tortured Kingdom* (TTK), is a novel first published serially in March 2026 and included as a contamination-resistant test set: because the four ChronoQA novels are old and famous, an answer LLM can produce correct answers from parametric memory alone, whereas a post-cutoff novel lets retrieval contribution be separated from memorisation. Its 101 questions were generated with Claude Opus 4.7 from ChronoQA’s published template (extended to all eight categories) and read for coherence, but not fact-checked against the source.

Methods. Five methods are compared under identical chunking (2000/400): pureLLM (no retrieval, raw question, clean parametric baseline); LightRAG (Guo et al., 2025) in hybrid mode at its default 30k-token budget; LightRAG-15K, the same capped at 15k; vanilla E²RAG (Zhang et al., 2026); and eSoKG. LightRAG and E²RAG return graph content *plus* the source chunks backing re-

Book	Src. tok.	#Q	Q words
Pride & Prejudice (P&P)	164k	54	36.5
The Phantom of the Opera	116k	70	26.0
Harry Potter (HP)	112k	25	30.0
The Tortured Kingdom (TTK)	90k	101	63.2
The Wonderful Wizard of Oz	50k	82	23.2
Total	—	332	—

Table 2: Evaluation corpus. Source length in c1100k tokens; mean question length in words. HP is *Harry Potter and the Philosopher’s Stone*; TTK is the post-cutoff novel.

trieved nodes; eSoKG is the only method returning graph content with no chunks. E²RAG’s combined (HyDE) mode is excluded because, for these famous books, the ungrounded hypothetical answer carries ground-truth vocabulary into retrieval, testing memorisation rather than the graph. All three graph methods and their retrievers live in one forked LightRAG codebase.

Models and protocol. Figure 1 summarises the evaluation. One LLM, Claude Sonnet 4.6, serves as indexer, answerer, judge, and the comparators’ retrieval-time keyword/cue extractor; one embedding model, Amazon Titan v2 (1024-d), is used everywhere. Sharing models removes model identity as a confound. The graph is built once per (book, method) and reused. Each (book, method, question) is answered five times, since the decoder runs at non-zero temperature. All methods share one answer prompt, adapted from LightRAG’s with the output-format constraints removed so that uncited methods are not penalised, and with external knowledge forbidden. An LLM judge scores each answer on a 1–10 rubric from the question, prediction, and reference only. Per-question scores are the mean over generations. Token cost is tracked per stage, attributing in-retriever LLM calls to the retrieval phase; eSoKG’s retrieval phase costs zero tokens. We report both the *all-questions* set (332) and the *common clean* set (260 questions every method answered with no refusal on any run). The judge scores on the rubric anchors of Table 3, returning an integer and a short rationale. The full prompts (QA generation, entity/event extraction, answer, and judge) are reproduced in Appendix E. Because the judge shares a model with the indexer and answerer, every answer was also re-scored by two judges from other model families, Qwen3-32B (Yang et al., 2025a) and DeepSeek-V3.2 (DeepSeek-AI, 2024), under the same prompt

Score	Anchor
10	Matches ground truth exactly or faithful paraphrase; all key points present.
7	Mostly correct; minor omissions or wording that do not change meaning.
5	Partially correct; some key points but major gaps or inaccuracies.
3	Mostly incorrect; only superficial overlap.
1	Off-topic, wrong, or hallucinated.

Table 3: Judge rubric anchors (1–10).

and rubric at temperature 0. Nothing was regenerated, so only the judge changes.

5 Results

Overall (Table 4). Under the primary judge, Sonnet 4.6, the two averaging schemes agree on the ranking. The four retrieval methods lie within a 0.6-point band, and the no-retrieval pureLLM is over two points below the lowest of them by-Q. eSoKG is second on the pooled set (7.21 by-Q; 7.79 clean), above E²RAG (7.12; 7.69) and the budget-matched LightRAG-15K (6.73; 7.50), and behind only the full-budget LightRAG (7.33; 7.92), which spends roughly twice the per-query tokens (Table 7). It also refuses on the fewest questions of any retrieval method (8.1%). Paired tests over the 332 per-question means (Table 5) make the margins over LightRAG-15K and pureLLM significant and leave the gap to LightRAG and the edge over E²RAG within noise. How this ordering holds up under two other judges is taken up below (*Independent judges*).

Refusals and the clean set. The shared prompt lets a method emit a fixed refusal string when its context is insufficient; pureLLM, which never receives that instruction, never refuses. Removing the union of all methods’ refusals leaves the 260-question common clean set, on which the ranking is unchanged (Table 4, *Clean*). Every method gains once refusal-prone questions are removed, but unevenly: LightRAG-15K gains most (+0.77, consistent with its 14.8% refusal rate), pureLLM least (+0.35, no refusals). eSoKG refuses least among retrieval methods and gains no more than LightRAG and E²RAG do from removing refusals (+0.58 against +0.59 and +0.57), so its standing against those two is not an artefact of abstention. Against LightRAG-15K, which refuses on 21 of 54 P&P and 19 of 101 TTK questions (Table A6), it partly is: eSoKG’s margin shrinks from +0.48 to

Method	Per-book mean (all-questions)					Average		Clean	Refuse
	HP	P&P	Phantom	Oz	TTK	by-Q	by-book	mean	(%)
LightRAG	8.35	6.07	7.94	7.13	7.48	7.33	7.39	7.92	9.0
eSoKG	8.20	6.81	8.01	7.06	6.74	7.21	7.36	7.79	8.1
E ² RAG	7.74	7.00	7.81	6.40	7.16	7.12	7.22	7.69	9.6
LightRAG-15K	8.26	5.05	7.67	6.80	6.53	6.73	6.86	7.50	14.8
pureLLM	7.34	5.47	6.13	4.94	2.47	4.70	5.27	5.05	0.0

Table 4: Mean score (1–10), Sonnet 4.6 judge (other judges: Table 5). *by-Q* averages over all 332 questions; *by-book* weights each book equally; *Clean mean* is over the 260-question common clean set; *Refuse* is the fraction of questions the method refused at least once. The four retrieval methods lie within a 0.6-point band; pureLLM is >2 points below the lowest by-Q. Best per column in bold.

	Sonnet 4.6	Qwen3-32B	DeepSeek-V3.2
<i>Mean of per-question means (rank)</i>			
LightRAG	7.33 (1)	7.79 (1)	8.18 (1)
eSoKG	7.21 (2)	7.54 (3)	8.15 (2)
E ² RAG	7.12 (3)	7.63 (2)	7.90 (3)
LightRAG-15K	6.73 (4)	7.27 (4)	7.35 (4)
pureLLM	4.70 (5)	5.01 (5)	4.80 (5)
<i>eSoKG – method, mean [95% CI]</i>			
LightRAG	−0.12 [−0.32, +0.08]	−0.25 [−0.46, −0.03]*	−0.04 [−0.30, +0.22]
E ² RAG	+0.08 [−0.13, +0.30]	−0.09 [−0.32, +0.14]	+0.25 [−0.03, +0.53]
LightRAG-15K	+0.48 [+0.26, +0.71]*	+0.27 [+0.02, +0.52]	+0.80 [+0.50, +1.11]*
pureLLM	+2.50 [+2.20, +2.81]*	+2.53 [+2.16, +2.90]*	+3.35 [+2.95, +3.74]*

Table 5: Judge robustness: the original and two independent judges scoring the *same* 8,300 answers. Top: mean over the 332 per-question means (rank). Bottom: paired difference eSoKG minus method, 95% bootstrap CI; *Holm-adjusted Wilcoxon $p < 0.05$. Agreement and per-book values in Appendix D.

+0.29 on the clean set.

Independent judges (Table 5). All 8,300 answers were re-scored under the same prompt by Qwen3-32B and DeepSeek-V3.2. Over the 332 per-question means both correlate closely with Sonnet 4.6 (Pearson $r = 0.92$ and 0.94 , Spearman $\rho = 0.85$ and 0.91 ; Table A4) and reproduce the large effects: pureLLM stays more than two points below every retrieval method, and LightRAG, eSoKG, and E²RAG lie within 0.3 points of one another under every judge. Within that band the judges differ. DeepSeek gives Sonnet 4.6’s ordering and the same four verdicts. Qwen places E²RAG above eSoKG (−0.09, not significant), makes eSoKG’s deficit to LightRAG significant (−0.25), and shrinks its margin over LightRAG-15K from +0.48 to +0.27. That margin’s bootstrap CI still excludes zero, but the Holm-adjusted Wilcoxon test, our criterion for significance, does not reject ($p = 0.63$ after Holm adjustment), so under Qwen we count it as not significant. The eSoKG–E²RAG swap under Qwen is a weighting effect. Under every judge eSoKG beats E²RAG on HP and Oz and loses on TTK, and TTK holds 101 of the 332 ques-

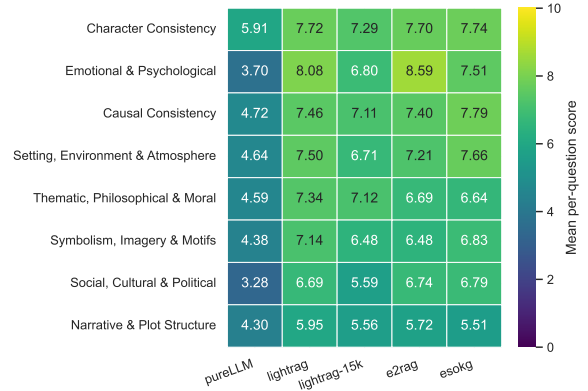


Figure 2: Mean score by category (rows, sorted by cross-method mean) and method, all books pooled, Sonnet 4.6 judge. eSoKG leads the retrieval methods on the consistency-style categories; every retrieval method is weakest on Narrative & Plot Structure.

tions, so the pooled by-Q mean leans on it; averaged by book, eSoKG leads E²RAG under Sonnet 4.6 and DeepSeek and ties under Qwen (Table A4). The ordering among the top three therefore depends on the judge and on the weighting. What holds under every judge is that eSoKG lies within 0.3 points of the best method.

By question type (Fig. 2). The per-category pattern fits the event-centric design. On the pooled set eSoKG leads the retrieval methods on Causal Consistency (7.79) and Character Consistency (7.74), the categories that ask how events and character state cohere across the book; the Causal lead holds under all three judges, the Character lead under two (Table A5). E²RAG leads on Emotional & Psychological, which rewards local event detail, and LightRAG on Thematic and Symbolism, which reward broad topical neighbourhoods. Every retrieval method scores lowest on Narrative & Plot Structure, where whole-book plot reasoning is hard whatever the graph.

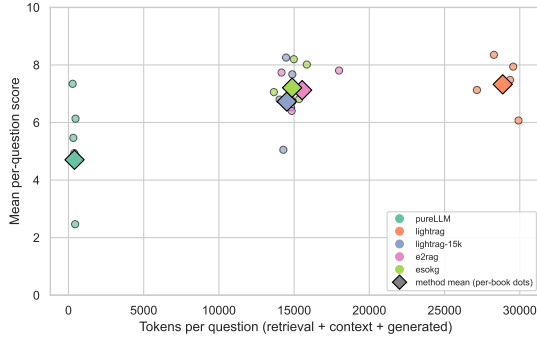


Figure 3: Quality vs. operational cost: mean judge score against mean total tokens per query. Small markers are (book, method) pairs; large diamonds are method means. eSoKG, E²RAG, and LightRAG-15K form a low-cost cluster at similar quality; LightRAG buys a small edge at almost twice the tokens.

Cost (Table 7, Fig. 3). Per query, eSoKG sits in the low-cost cluster (14.9k tokens), close to LightRAG-15K (14.5k) and E²RAG (15.5k) and about half of full-budget LightRAG (28.9k). What is distinctive is the *shape* of the cost: eSoKG is the only method with zero retrieval-stage tokens, because it retrieves via embeddings and a graph walk with no pre-retrieval LLM call, and spends its budget on the answer instead (it writes the longest answers on every book, ~20% longer than LightRAG). Within the low-cost cluster it has the highest pooled mean under Sonnet 4.6 and DeepSeek, and the second highest, 0.09 behind E²RAG, under Qwen (Table 5). The drawback is the build cost. Building the HP graph cost eSoKG 3.27M tokens against 1.34M (LightRAG) and 1.12M (E²RAG), split between QA-mediated extraction (2.12M) and deduplication (1.15M). Against full-budget LightRAG the per-query saving overtakes this build premium only after ~140 queries on this book; against the cheaper LightRAG-15K it never does. Table 6 decomposes the build cost; per-book graph sizes are in Appendix B, where eSoKG’s merging is visible as far fewer entity nodes than the unmerged E²RAG (438 vs. 1,592 on HP).

Post-cutoff novel. TTK is the one book the models are very unlikely to have seen in pre-training. pureLLM collapses from 7.34 on HP to 2.47 here while every retrieval method holds between 6.53 and 7.48 (Table 4), so retrieval contributes beyond memory, and the other books’ scores partly reflect recall. TTK shows no eSoKG advantage in overall score. eSoKG (6.74) sits behind LightRAG (7.48)

Method	Calls	Prompt tok	Total tok
LightRAG	206	1.09M	1.34M
E ² RAG	142	0.79M	1.12M
eSoKG	474	2.18M	3.27M
<i>Ph.1 (extract)</i>	220	1.23M	2.12M
<i>Ph.2 (dedup)</i>	254	0.94M	1.15M

Table 6: One-time construction cost for *Harry Potter* (2000/400 chunking). eSoKG is the most expensive to build; the two indented rows decompose it.

Method	Retr.	Ctx.	Gen.	Total
pureLLM	0	0	402	402
LightRAG	808	27,470	590	28,867
LightRAG-15K	808	13,186	522	14,516
E ² RAG	516	14,476	542	15,534
eSoKG	0	14,169	698	14,867

Table 7: Mean per-query tokens by stage (over all books). **Retr.:** pre-retrieval keyword/cue LLM call. **Ctx.:** retrieved context. **Gen.:** answer tokens. eSoKG pays nothing at retrieval. Ctx. counted with c1100k; Retr. and Gen. as reported by the API. LightRAG-15K shares LightRAG’s retrieval call.

and E²RAG (7.16) and above LightRAG-15K (6.53) under all three judges, and it leads no category (Table A7). The shape of its scores across categories is nevertheless close to the pooled one. Setting and Character remain its two strongest categories, and Narrative, Thematic, and Symbolism its three weakest. The one change is Causal Consistency, its best category on the pooled set and fifth of eight here. Since this profile appears on a book the model has not memorised, it more likely comes from the method than from recall.

Qualitative example. Figure 4 shows what each graph retrieves for one question: “*who gave Hagrid the dragon egg in the Hog’s Head, and why was it significant?*” Answering it means linking the pub exchange to Hagrid’s later disclosure of how to get past Fluffy. eSoKG retrieves both events as named, connected nodes, with Quirrell, Harry, and Ron each a single merged node. E²RAG, which does not merge, leaves Hagrid, the egg, and Fluffy spread over several nodes and surfaces the dragon’s aftermath instead of the pub exchange. LightRAG returns a broad topical neighbourhood with no event nodes and with mentions left apart (*Quirrell* and *Professor Quirrell* are separate nodes). eSoKG’s merging is imperfect too: *Hagrid* and *Dumbledore* each appear as two nodes here because a late description drifted past the embedding threshold, the

case a name check would catch (Appendices C and F).

6 Discussion

RQ1 (quality): yes on the pooled set, not on the post-cutoff book. On the 332 pooled questions a context with *no source text* scores within 0.3 points of the two methods that hand the model the passages, under every judge, and neither gap is significant under Sonnet 4.6 (Qwen makes the gap to LightRAG significant, -0.25). For these books the distilled structure carries enough to answer. On TTK it does not match: eSoKG trails both graph comparators there, so part of the pooled parity reflects books the model already knows. The result is also bounded by the setup. The same model wrote the structure and answered from it, so what is shown is that the structure preserves enough *for that model*, not that the graph is complete on its own.

Why the gap to LightRAG is small. LightRAG’s edge, 0.12 points under Sonnet 4.6 and 0.04 to 0.25 under the independent judges, tracks its budget: it feeds $\sim 27.5k$ context tokens to eSoKG’s 14.2k, and at a matched budget LightRAG-15K falls *below* eSoKG under all three judges, significantly under two. What the budget buys is mostly coverage. The two books where LightRAG-15K refuses most, P&P and TTK, are the two where LightRAG beats it by about a point under every judge. On HP and Phantom, where refusals are rare, the gap closes, and under Qwen LightRAG-15K scores above every other method on both books (Table A4). eSoKG’s margin over LightRAG-15K is likewise concentrated on P&P, and the Qwen result shows what kind of margin it is: the mean difference stays positive because refusals score near the floor, while the Wilcoxon signed-rank test, which is insensitive to a tail of large differences, finds the median per-question difference indistinguishable from zero. The matched budget also shows a coverage effect in eSoKG’s favour. The same 15k tokens are enough to answer more often when they hold events than when they hold entity summaries (8.1% refusals against 14.8%), and where both methods answer, eSoKG still leads under Sonnet 4.6 ($+0.29$ on the clean set) and DeepSeek, and ties under Qwen (Appendix D). The event-centric context therefore covers more of what the questions ask for per token; whether it also answers better where both have enough, two of three judges say it does and one

says it does not. eSoKG’s longer answers remain an untested confound: length sensitivity is not measured here, and the one published result for the judge’s model family finds a conciseness rather than a verbosity preference (Soumik, 2026). That a model relies more on its context when the context gives it more evidence is consistent with (Sun et al., 2026).

RQ2 (cost): competitive per query, costly to build. Per query eSoKG is parity-or-better against the cheap methods and a clear saving only against the full-budget one; its real premium is at build time, where the QA-mediated extraction and the cross-chunk merging together cost $2\text{--}3\times$ the comparators’ tokens. The per-query saving offsets this only slowly and only against the full-budget comparator (§5), so as a drop-in retrieval index for one-shot QA the build cost is hard to justify. Whether it is worth paying depends less on query volume than on what the graph is *used* for, which the next point takes up.

What the structure supports beyond QA. The evaluation measures one use of the graph, but the merged event-centric structure supports operations the comparators cannot serve directly. Because events are first-class nodes and merging consolidates each entity into one, and because the chunk-free design makes the structure *the graph* rather than an index into text, several useful queries are short graph walks needing no language model: the events two characters share (the overlap of their event neighbourhoods), what led to and followed an event (its incoming and outgoing edges), and a character’s arc (the ordered chain of events they take part in). LightRAG has no event nodes to walk between; E²RAG has them but leaves the same character and event split across unmerged copies, so any such query must first regather the fragments, which is the work eSoKG’s merging already did. These are properties of the representation, not measured results. They are, however, the operations a graph-navigating agent (Ghandi et al., 2026; Ni et al., 2025) would compose, and the one-shot RAG setting of our evaluation exercises none of them, which is probably the setting that least favours a traversable graph.

7 Conclusion

We asked whether an event-centric KG built with 5W1H QA-mediated extraction and cross-chunk

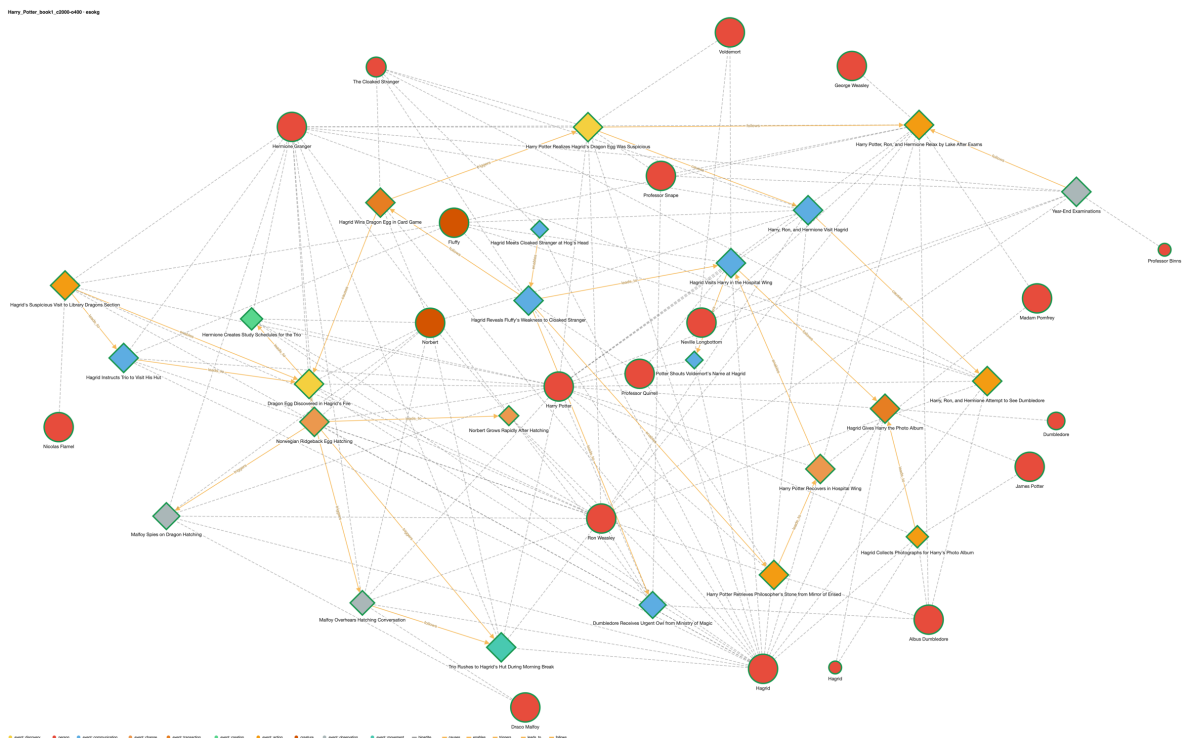


Figure 4: Subgraph eSoKG retrieves for the query “*who gave Hagrid the dragon egg in the Hog’s Head, and why was it significant?*” Circles are entities, diamonds are events (coloured by subtype), and edges are typed participation and causal/temporal links (*causes*, *enables*, *triggers*, *leads_to*, *follows*). The two events the question turns on, *Hagrid Meets Cloaked Stranger at Hog’s Head* and *Hagrid Reveals Fluffy’s Weakness to Cloaked Stranger*, are both retrieved and connected, and major characters (Harry, Quirrell, Ron) appear as single merged nodes. Merging is imperfect: *Hagrid* and *Dumbledore* each appear here as two nodes (Appendix C gives whole-graph counts).

merging by description, queried chunk-free, can answer narrative questions as well as existing graph-based retrieval (RQ1), and at what cost (RQ2). On quality, the pooled evaluation places eSoKG within half a point of an unmerged entity–event comparator and of a full-budget entity-centric one that spends twice the per-query tokens, under the original and two independent judges, with no source text shown to the model. What is statistically robust is narrower than that. Under every judge eSoKG beats the no-retrieval baseline by more than two points, and under two of three it beats LightRAG-15K, the entity-centric method at a matched budget. Its gap to LightRAG and its edge over E²RAG are within noise under Sonnet 4.6 and DeepSeek; under Qwen the gap to LightRAG becomes significant and E²RAG moves ahead by a non-significant margin. The ordering among eSoKG, E²RAG, and LightRAG therefore depends on the judge. Merging mentions by description leaves the graph able to answer causal-consistency questions, where eSoKG leads under every judge, and character-consistency questions, where it leads under two of three, which

is the opposite of what the closest prior work assumes. The parity does not extend to the one post-cutoff book, where eSoKG trails both graph comparators, so part of it reflects books the model already knows. On cost, the saving is per query and the price is a one-time build two to three times the comparators’. The merged event-centric graph also answers character-arc, shared-event, and causal-chain queries as direct graph walks, which the comparators cannot serve directly and which the one-shot evaluation does not exercise. For event-centric narrative KGs, description-based merging is a workable alternative to keeping mentions separate on the books tested here. Confirming it needs several post-cutoff novels and a judge closer to a human reader.

Limitations

The evaluation is a whole-system comparison. eSoKG differs from the comparators in four coupled ways (event schema, 5W1H extraction, description merging, chunk-free context) and is not internally ablated, so no result can be credited to one choice

alone. Separating them is the natural next step.

The primary judge is the same model that builds the graphs and writes the answers, and it has read most of the books, so absolute scores partly reflect recall and small gaps are unreliable. The pooled ranking is stable across the all-questions and clean views, and pureLLM’s collapse on the post-cutoff book shows the judge punishes ungrounded answers. Re-scoring every answer with two independent judges (Table 5) reproduces the large effects, but the two disagree on the fine ordering. One reproduces Sonnet 4.6’s ranking and all four verdicts. The other swaps eSoKG and E²RAG, makes the deficit to LightRAG significant, and leaves the margin over LightRAG-15K positive in mean but not under the signed-rank test. The ordering among retrieval methods is therefore judge-dependent, and no LLM judge here can show which is closer to a human reader. eSoKG also writes the longest answers, and the judges’ sensitivity to answer length is not measured.

Pooled means fold refusals, scored near the floor, into answer quality, and weight TTK’s 101 questions four times HP’s 25. The clean set and the by-book average (Table 4) separate these effects, but the paired tests inherit the pooled weighting.

On the one contamination-free book eSoKG trails both graph comparators, so the pooled parity partly reflects books the model already knows. A firmer claim needs several post-cutoff novels. TTK’s questions are also longer than the ChronoQA books’ (63.2 vs. 23–37 words) and were generated by a different model from an extended template, so the HP–TTK contrast varies question style as well as contamination.

All questions and reference answers are LLM-generated and not human-verified, so the evaluation measures agreement with an LLM-written reference. Construction-cost figures come from one book. The cross-chunk merging is not guaranteed to succeed: a character whose mentions fail to merge is split over several nodes, and the evaluation does not measure how often this happens or what it costs.

Acknowledgements

The author thanks Ricky Molén for insightful discussions on the design and evaluation, and the anonymous reviewers for their detailed and constructive feedback. The author used Claude Code for language polishing, code assistance, and anal-

ysis scripting; all technical content, experimental results, and conclusions were verified by the author.

References

- Jiaxin Bai, Wei Fan, Qi Hu, Qing Zong, Chunyang Li, Hong Ting Tsang, Hongyu Luo, Yauwai Yim, Haoyu Huang, Xiao Zhou, Feng Qin, Tianshi Zheng, Xi Peng, Xin Yao, Huiwen Yang, Leijie Wu, Yi Ji, Gong Zhang, Renhai Chen, and Yangqiu Song. 2025. [AutoSchemaKG: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora](#).
- Sanghyeok Choi, Woosang Jeon, Kyuseok Yang, and Taehyeong Kim. 2026. [SocraticKG: Knowledge graph construction via QA-driven fact extraction](#).
- Peter. Christen. 2012. *Data matching : concepts and techniques for record linkage, entity resolution, and duplicate detection*, 1st ed. 2012. edition. Data-centric systems and applications. Springer-Verlag.
- DeepSeek-AI. 2024. [DeepSeek-V3 technical report](#). *arXiv preprint arXiv:2412.19437*.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. [From local to global: A graph RAG approach to query-focused summarization](#).
- Taraneh Ghandi, Hamidreza Mahyar, and Shachar Klaiman. 2026. [GraphWalk: Enabling reasoning in large language models through tool-based graph navigation](#). *Preprint*, arxiv:2604.01610 [cs.AI].
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. [LightRAG: Simple and fast retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. Association for Computational Linguistics.
- Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Luu Anh Tuan. 2025. [HyperGraphRAG: Retrieval-augmented generation via hypergraph-structured knowledge representation](#).
- Tengjun Ni, Xin Yuan, Shenghong Li, Kai Wu, Ren Ping Liu, Wei Ni, and Wenjie Zhang. 2025. [StepChain GraphRAG: Reasoning over knowledge graphs for multi-hop question answering](#). *Preprint*, arxiv:2510.02827 [cs.CL].
- Sadman Kabir Soumik. 2026. [Judging the judges: A systematic evaluation of bias mitigation strategies in LLM-as-a-judge pipelines](#).
- Kaiser Sun, Fan Bai, and Mark Dredze. 2026. [Task matters: Knowledge requirements shape LLM responses to context-memory conflict](#).

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. [ChatIE: Zero-shot information extraction via chatting with ChatGPT](#). *Preprint*, arxiv:2302.10205 [cs.CL].

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. [On the representation and embedding of knowledge bases beyond binary relations](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1300–1307.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.

Zairun Yang, Yilin Wang, Zhengyan Shi, Yuan Yao, Lei Liang, Keyan Ding, Emine Yilmaz, Huajun Chen, and Qiang Zhang. 2025b. [EventRAG: Enhancing LLM generation with event knowledge graphs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16967–16979. Association for Computational Linguistics.

Ze Yu Zhang, Zitao Li, Yaliang Li, Bolin Ding, and Bryan Kian Hsiang Low. 2026. [Respecting temporal-causal consistency: Entity-event knowledge graph for retrieval-augmented generation](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2017–2054, Rabat, Morocco. Association for Computational Linguistics.

A Implementation Defaults

Table A1 lists the configuration used by the eSoKG pipeline (§3.2). These are configuration, not tuned hyperparameters: they were fixed by light iteration during development on a held-out short story (Liam O’Flaherty’s *The Sniper*, used nowhere in the evaluation) and were not re-tuned on the evaluation corpus.

B Graph Statistics

Table A2 reports the size and shape of each method’s graph per book, after deduplication. Three structural differences are explicit in the counts: eSoKG has *no* entity–entity edges (entities relate only through shared events); LightRAG has no event nodes or bipartite edges (it is entity-centric); and eSoKG has far fewer entity nodes than the unmerged E²RAG on every book, a direct consequence of its cross-chunk merging.

Parameter	Default
Entity dedup thresholds (pass 1 & 2)	0.8, 0.7, 0.6, 0.5
Event dedup thresholds (pass 1 & 2)	0.85, 0.75, 0.65
Max / min candidates per LLM dedup call	20 / 15
Concurrent dedup calls	4
Max extraction output tokens	16384
Per-chunk retries	3
Chunk size / overlap (eval)	2000 / 400
Retrieval seeds per path (k)	10
Expansion depth (hops)	2
Retrieval token budget	15k

Table A1: eSoKG configuration defaults. Event thresholds are kept above entity thresholds because a false event merge is unrecoverable.

Book	Method	Ent.	Evt.	E–E	Ev–Ev	Bip.
HP	eSoKG	438	770	0	808	4,629
	LightRAG	706	—	1,167	—	—
	E ² RAG	1,592	1,310	1,197	860	3,819
TTK	eSoKG	378	716	0	773	3,648
	LightRAG	544	—	843	—	—
	E ² RAG	972	1,219	818	845	2,922
P&P	eSoKG	264	882	0	934	5,050
	LightRAG	524	—	1,040	—	—
	E ² RAG	1,787	1,650	1,401	1,072	4,696
Oz	eSoKG	144	335	0	355	1,891
	LightRAG	229	—	390	—	—
	E ² RAG	548	649	496	487	1,885
Phantom	eSoKG	421	686	0	757	3,472
	LightRAG	649	—	1,009	—	—
	E ² RAG	1,147	1,156	969	791	2,956

Table A2: Node and edge counts per book and method after deduplication. **Ent./Evt.**: entity/event nodes. **E–E**, **Ev–Ev**, **Bip.**: entity–entity, event–event, bipartite edges. A dash means the schema has no such element.

C Deduplication on *Harry Potter*

The *Harry Potter* graph has 438 entity nodes after Phase 2. Table A3 reports, per principal character, how many person nodes survive, the degree of the dominant node, and the degrees of any stragglers. Five of ten characters consolidate to a single node; the rest keep a dominant node plus one to three stragglers whose degrees are in the single digits against a dominant node of 63 or more. Fragmentation, where it happens, is minor in weight, and the missed merges are exactly the cases a name check would catch (the stragglers share the dominant node’s surface name). LightRAG, which merges by name, fails the opposite way: on the same book its graph keeps *Quirrell* and *Professor Quirrell* as separate nodes where eSoKG consolidates both. The two failure modes are mirror images, motivating

the hybrid name+description criterion suggested in §6. These are illustrative examples, not a measured merge-accuracy comparison.

Character	Nodes	Dom. deg.	Straggler deg.
Harry Potter	1	611	—
Ron Weasley	4	294	6, 4, 2
Hermione Granger	4	202	3, 2, 1
Hagrid	3	189	3, 0
Albus Dumbledore	2	102	5
Professor Snape	1	88	—
Neville Longbottom	1	87	—
Voldemort	1	80	—
Draco Malfoy	2	77	2
Professor Quirrell	1	63	—

Table A3: Surviving person nodes per principal character after Phase 2, ordered by dominant-node degree. A dash means the character consolidated to a single node.

D Supplementary Results

How scores are computed. Every score in this paper is a mean of per-question means. Each (book, method, question) triple is answered five times; a judge scores each of the five answers once on the 1–10 rubric of Table 3; the five scores are averaged into a per-question mean; and a table cell is the mean of those per-question means over the questions in its scope (a book, a category, the 332-question pooled set, or the 260-question clean set). A table that does not name a judge reports the primary judge, Sonnet 4.6. The two independent judges, Qwen3-32B and DeepSeek-V3.2, scored the same 8,300 answers and appear only where a caption names them. Refusals are detected from the answer text and do not depend on any judge.

D.1 Independent judges

Table A4 gives the per-book means under the two independent judges behind Table 5, with per-book agreement with Sonnet 4.6. On the 260-question clean set the orderings are: Qwen3-32B LightRAG 8.39, E²RAG 8.19, LightRAG-15K 8.18, eSoKG 8.12, pureLLM 5.45; DeepSeek-V3.2 eSoKG 8.96, LightRAG 8.95, E²RAG 8.65, LightRAG-15K 8.44, pureLLM 5.32. Table A5 gives the per-category means under all three judges.

D.2 Refusals

Table A6 gives, per book, the questions on which each retrieval method refused at least once; removing their union leaves the 260-question common clean set used in §5.

	HP	P&P	Phant.	Oz	TTK	by-book
<i>Qwen3-32B</i>						
LightRAG	8.82	6.59	8.11	7.54	8.15	7.84
eSoKG	8.66	6.91	7.95	7.41	7.42	7.67
E ² RAG	8.14	7.26	8.08	7.00	7.89	7.67
LightRAG-15K	8.84	5.55	8.17	7.30	7.15	7.40
pureLLM	7.72	5.66	6.78	5.76	2.16	5.62
r / ρ vs. Sonnet 4.6	.76 / .68	.93 / .89	.82 / .68	.93 / .90	.94 / .89	.92 / .85
Offset vs. Sonnet 4.6	+0.46	+0.31	+0.30	+0.54	+0.48	+0.43
<i>DeepSeek-V3.2</i>						
LightRAG	9.41	6.54	9.07	7.84	8.42	8.26
eSoKG	9.41	7.54	9.29	7.87	7.59	8.34
E ² RAG	8.66	7.37	9.00	7.03	7.92	8.00
LightRAG-15K	9.34	5.00	8.79	7.51	6.97	7.52
pureLLM	8.10	4.90	6.74	5.46	2.05	5.45
r / ρ vs. Sonnet 4.6	.85 / .72	.94 / .93	.91 / .84	.95 / .95	.93 / .91	.94 / .91
Offset vs. Sonnet 4.6	+1.01	+0.19	+1.07	+0.68	+0.52	+0.66

Table A4: Independent judges, all-questions set: per-book mean of per-question means (best per column in bold within each judge), Pearson r / Spearman ρ with Sonnet 4.6 over per-question means, and mean offset; the last column pools all 332 questions. By category, all three judges give eSoKG the lead on Causal and Setting; Character and Social hold under DeepSeek and pass to LightRAG under Qwen. Neither judge ever returned 2, 4, or 6, and both returned valid JSON on all 8,300 calls.

D.3 Post-cutoff book

Table A7 gives the per-category breakdown on the post-cutoff *The Tortured Kingdom* under Sonnet 4.6, where pureLLM collapses across the board while the retrieval methods hold.

E Prompts

This appendix reproduces the prompts as run; the worked example in the event-extraction prompt has been lightly edited for the paper. Braced tokens (e.g. {chunk_text}, {context}) are placeholders filled at runtime.

E.1 Question generation (post-cutoff set)

The question set for *The Tortured Kingdom* was generated with Claude Opus 4.7 from the template below, extended from ChronoQA’s published template (which ships only a Causal-Consistency worked example) to cover all eight categories, one generation pass per category. {{DOCUMENT}} is replaced with the source text and {{CATEGORY_BLOCK}} with one block per pass.

Question-generation master template

I want to test a few rag systems on their reasoning capabilities and the capability to pick up nuanced details. Use the document shown below, design queries for it, also provide the ground truth for each query. In addition, it should focus on testing the RAG system’s causal consistency. for example, 1. it can ask the RAG system how would a certain character behavior in a given context/even from the document, and see if the answer is consistent with the character’s personality/traits at that specific

Category	Sonnet 4.6				Qwen3-32B				DeepSeek-V3.2			
	LR	eSoKG	E ²	LR15	LR	eSoKG	E ²	LR15	LR	eSoKG	E ²	LR15
Causal Consistency	7.46	7.79	7.40	7.11	8.06	8.26	7.98	7.99	8.53	9.05	8.19	7.86
Character Consistency	7.72	7.74	7.70	7.29	8.27	8.05	8.12	7.77	8.63	8.73	8.42	7.84
Emotional & Psychological	8.08	7.51	8.59	6.80	8.45	7.95	9.18	7.28	8.58	8.18	9.53	7.28
Narrative & Plot Structure	5.95	5.51	5.72	5.56	6.33	5.67	5.98	6.02	6.92	6.59	6.41	6.21
Setting, Env. & Atmosphere	7.50	7.66	7.21	6.71	8.02	8.28	7.96	7.37	8.29	8.51	8.11	7.04
Social, Cultural & Political	6.69	6.79	6.74	5.59	7.36	7.17	7.25	6.21	7.32	7.61	7.44	6.05
Symbolism, Imagery & Motifs	7.14	6.83	6.48	6.48	7.43	7.08	7.13	6.99	7.99	7.76	7.24	7.30
Thematic, Phil. & Moral	7.34	6.64	6.69	7.12	7.57	6.64	6.81	7.28	8.23	7.32	7.42	7.95

Table A5: Per-category mean of per-question means for the four retrieval methods under each judge, all books pooled (LR=LightRAG, E²=E²RAG, LR15=LightRAG-15K); best retrieval method per category and judge in bold. eSoKG leads Causal and Setting under every judge, Character and Social under Sonnet 4.6 and DeepSeek; E²RAG leads Emotional and LightRAG leads Narrative, Symbolism, and Thematic under every judge. pureLLM (not shown) is lowest in every cell.

Method	HP	P&P	Phantom	Oz	TTK	Total
LightRAG	0	13	0	9	8	30
eSoKG	1	8	0	4	14	27
E ² RAG	2	9	0	7	14	32
LightRAG-15K	0	21	0	9	19	49
Questions	25	54	70	82	101	332
In clean set	23	31	70	69	67	260

Table A6: Questions on which each retrieval method refused at least once, by book, with questions per book and questions remaining in the common clean set. pureLLM never refuses. Refusals are detected from the answer text and do not depend on the judge.

Category	pLLM	LR	LR15	E ² RAG	eSoKG
Char.	1.84	7.06	5.98	7.60	7.48
Causal	2.32	7.46	6.91	6.98	6.96
Narr.	3.02	6.50	5.52	5.84	4.90
Set.	1.89	7.80	6.68	7.60	7.67
Social	1.35	7.75	6.03	7.34	7.03
Emot.	1.35	8.06	7.20	8.18	7.17
Sym.	3.10	6.78	5.44	6.04	5.90
Theme	5.96	8.00	7.96	7.26	5.90
Overall	2.47	7.48	6.53	7.16	6.74

Table A7: Per-category mean score on the post-cutoff *The Tortured Kingdom*, all questions, Sonnet 4.6 judge. pLLM=pureLLM, LR=LightRAG, LR15=LightRAG-15K; best per row in bold.

point in time (character can experience development so their response to the same thing can vary). 2. it can test if the RAG system confuse the event that has not happened but in the document with the query (the future events that has not happen should not be account for in the answer). Give the queries and ground truth in json format. The document is here:

{{DOCUMENT}}

Focus on Causal Consistency: Apart from character behavior over time and future event confusion, other causal consistency tests (e.g., testing how well it understands cause-and-effect relationships in the plot) can also be included. Query Difficulty: query should involve complex, nuanced reasoning/ understanding of the document provided.

In this run, generate questions ONLY for the following category:

{{CATEGORY_BLOCK}}

Each query object must have EXACTLY three fields: `query`, `ground\_truth`, and `type`.

Format example:

```
{
  "queries": [
    {
      "query": "During Harry's first night at Hogwarts, the Gryffindor students climb the moving staircases. How does the text describe the corridors and stairways' magical behavior, and what is the immediate impact on Harry's sense of direction?",
      "ground_truth": "The staircases sometimes change direction, doors can vanish or move, and some
```

require a password or a specific tickle of a doorknob to open. This constant shifting confuses new students like Harry, making it easy to get lost early on.",
"type": "Causal Consistency"

```
}
]
}
```

It is also important to note that do not explicitly disclose the title/chapter/section number from which the context of the question is used. Simply described the related event and the characters involved to make the context clear.

Category blocks (one per generation run)

Category: Character Consistency
Description: Questions probing character motivations, development, or psychology. Especially valuable: questions whose correct answer depends on WHERE in the story the reference point falls, because characters develop and a single character's "right" response can change between early and late story.
Set "type" field to: "Character Consistency"

Category: Causal Consistency
Description: Questions about cause and effect, logical sequences of events, or explanations of how events unfolded. Include some questions that test whether the system avoids leaking future events (events that have not yet happened relative to the query's reference point) into its answer.

Set "type" field to: "Causal Consistency"

Category: Symbolism, Imagery & Motifs
Description: Questions about symbolic elements, recurring imagery, or metaphorical representations in the story.
Set "type" field to: "Symbolism, Imagery & Motifs"

Category: Thematic, Philosophical & Moral
Description: Questions about deeper meanings, philosophical ideas, or ethical implications raised by the story.
Set "type" field to: "Thematic, Philosophical & Moral"

Category: Narrative & Plot Structure
Description: Questions about story organization, plot devices, or narrative techniques (e.g., framing, foreshadowing, point of view shifts).
Set "type" field to: "Narrative & Plot Structure"

Category: Setting, Environment & Atmosphere
Description: Questions about physical locations, time periods, or mood/atmosphere of specific scenes.
Set "type" field to: "Setting, Environment & Atmosphere"

Category: Social, Cultural & Political
Description: Questions about societal contexts, cultural elements, or political dimensions of the story.
Set "type" field to: "Social, Cultural & Political"

Category: Emotional & Psychological
Description: Questions about emotional responses, psychological states, or mental processes of characters at specific moments.
Set "type" field to: "Emotional & Psychological"

- These perspectives are **guides** to inspire multiple possible QA pairs, even if they are implicit or only partially expressed.

4. **Coverage**

- Extract all possible QA pairs that can be reasonably derived from the document.

OUTPUT FORMAT
Return a JSON list of QA objects:

```
[
  { "question": "...", "answer": "...",
    ...
  ]
```

Entity-extraction prompt (Step 2)

---Role---
You are a Named Entity Specialist responsible for identifying and describing entities from a set of document-grounded Question-Answer pairs.

---Context---
The QA pairs below were generated from a single chunk of a source document using 5W1H-guided analysis. Each QA pair is self-contained with pronouns resolved to explicit entity names.

---Entity Definition & Boundary Rules---

An entity is a specific, nameable person, place, organization, object, concept, or other distinct thing that participates in, is affected by, or is otherwise central to the events and facts described in the QA pairs.

****Inclusion criteria**** -- extract it if:

- It is a named or uniquely identifiable thing (proper noun, specific concept, named method, titled work, etc.).
- It plays an active role (agent, patient, instrument) in at least one event or fact in the QA pairs.
- It would warrant its own node in a knowledge graph extracted from these QA pairs.

****Exclusion criteria**** -- do NOT extract:

- Generic temporal expressions ("six months", "recently", "last year") unless they name a specific period (e.g., "Q3 2023", "The Great Depression").
- Bare quantities or metrics ("18%", "\$5 million", "three times") unless they are the name of something specific.
- Vague or generic noun phrases that do not refer to a distinct entity ("the situation", "the issue", "the problem", "the process").
- Adjectives, descriptors, or properties used in passing ("large-scale", "innovative").
- Common nouns used generically ("a company", "some researchers") rather than referring to a specific entity.

---Entity Type Definitions---
Classify each entity using exactly one of the following types. Use "Other" only if none fit.

- Person: A specific named individual (real or fictional).
- Creature: A specific named non-human living being or species discussed as a distinct entity.
- Organization: A named company, institution, agency, team, or formal group.
- Location: A named geographic place, region, facility, or address.
- Concept: A specific named idea, theory, framework, field, or principle that the text discusses substantively (not merely references in passing).
- Method: A specific named technique, algorithm, process, or methodology.
- Content: A named creative or informational work (article, book, report, dataset publication, speech, etc.).
- Data: A specific named dataset, database, corpus, or data collection.
- Artifact: A named physical or digital object, tool, system, product, or piece of infrastructure.

E.2 Extraction prompts (Phase 1)

The three sequential per-chunk calls of §3.2: QA generation (Step 1), entity extraction (Step 2), and event extraction with intra-chunk relations (Step 3).

QA-generation prompt (Step 1)

ROLE
You are a **Comprehensive Knowledge Archivist** who converts the [Full Document] into detailed, document-grounded QA pairs.

OBJECTIVE
Extract as many meaningful Question-Answer pairs as possible from the document.
Use the 5W1H perspectives (Who, What, When, Where, Why, How) **as analytical lenses** to help you identify and expand potential questions, but do NOT restrict yourself to producing only 5W1H-type questions.
Your goal is to maximize informational coverage, capturing every explicit fact, relation, event, definition, rationale, and process described in the document.

INPUT
Full Document: "{chunk_text}"

CONSTRAINTS

- **Context-Independent****
 - Each QA must be self-contained and understandable without referencing the original text.
 - Replace pronouns with explicit entities.
- **No Hallucination****
 - Use only facts explicitly stated in the document.
- **Expansion-Oriented Thinking****
 - For each sentence or factual unit, consider the 5W1H perspectives as prompts to explore:
 - WHO is involved?
 - WHAT happened or is described?
 - WHEN did it occur?
 - WHERE did it occur?
 - WHY did it occur?
 - HOW was it carried out?

- NaturalObject: A named natural entity (river, mountain, species, chemical compound, celestial body, etc.).

****Special guidance for Concept and Method types:****
Only extract a Concept or Method if it is specific and named enough to be a distinct knowledge graph node. "Machine learning" as a general field mentioned in passing: do not extract. "Retrieval-Augmented Generation" as a specific technique discussed in the text: extract. When in doubt, ask: "Does the text say something *about* this concept, or merely use it as a descriptor?"

---Canonicalization & Deduplication Rules---

If the same real-world entity is referred to by different surface forms within the QA pairs, unify them under a single canonical entry.

- ****Canonical name:**** Use the most complete, formal, or unambiguous version. Prefer full names over abbreviations (e.g., "World Health Organization" over "WHO"), but use the form most recognizable in context.
- ****Aliases:**** List ALL surface-form variants found in the QA pairs, including the canonical name itself.

****Specific merge/split rules:****

- ****Name variants:**** "Barack Obama", "Obama", "Barack Hussein Obama" -> merge. Canonical: "Barack Obama".
- ****Abbreviations:**** "WHO" and "World Health Organization" -> merge. List both as aliases.
- ****Role-based references:**** If the text uses a role ("the CEO") and a name ("Sarah Chen") to refer to the same individual*, merge them. Include the role-based form in aliases. However, if a role refers to the position generically (not a specific person), do not merge it with any person entity.
- ****Part-whole relationships:**** Keep as SEPARATE entities if they are genuinely distinct things that the text treats differently (e.g., "U.S. Senate" and "U.S. Congress" are separate if the text discusses them independently). Merge only if the text uses them interchangeably for the same referent.
- ****Never create two entries for the same real-world entity.****

---Instructions---

****Step 1: Entity Identification****
Read through ALL the QA pairs. Identify every distinct entity that meets the inclusion criteria above. Apply the exclusion rules strictly.

****Step 2: Cross-QA Synthesis****
If the same entity appears across multiple QA pairs, combine all information into a single entry. If different surface forms refer to the same entity, canonicalize per the rules above.

****Step 3: Entity Details****
For each entity, extract:

- **`entity\_id`:** A short stable identifier (EN1, EN2, EN3, ...).
- **`entity\_name`:** The canonical name. Use title case for proper nouns.
- **`entity\_type`:** One of the defined entity types above.
- **`aliases`:** A list of all surface-form variants found in the QA pairs (including the canonical name).
- **`description`:** 1-3 sentences (up to 80 words). Summarize this entity's role, key attributes, and relationships as described in the QA pairs. Write in third person using explicit entity names (never pronouns). Preserve important facts, names, dates, and relationships. Every word must carry information -- no filler.

****Step 4: Objectivity & Language****

- Write all descriptions in the third person. Use explicit entity names, never pronouns.
- Output must be in {language}. Retain proper nouns in their original language.

---Pre-Output Validation Checklist---

Before returning your output, verify all of the following:

- Every ``entity_type`` is one of the defined types above.
- No two entries refer to the same real-world entity (deduplication check).
- Every alias in the ``aliases`` list actually appears in the QA pairs.
- No entry is a generic noun phrase, bare metric, temporal expression, or passing descriptor.
- All descriptions use entity names, not pronouns.

---Worked Example---

****Example QA Pairs:****

```
[
  {"question": "What did Arcadia Corp announce in March 2023?", "answer": "Arcadia Corp announced the launch of the Greenfield Initiative in March 2023."},
  {"question": "Who led the Greenfield Initiative?", "answer": "Sarah Chen, VP of Sustainability at Arcadia Corp, led the Greenfield Initiative."},
  {"question": "Why did Arcadia Corp launch the Greenfield Initiative?", "answer": "Arcadia Corp launched the Greenfield Initiative in response to new environmental regulations imposed by the City of Millbrook in January 2023."},
  {"question": "What happened after the Greenfield Initiative launched?", "answer": "After the Greenfield Initiative launched, Arcadia Corp reduced carbon emissions by 18 percent within six months."},
  {"question": "What framework did the Greenfield Initiative use?", "answer": "The Greenfield Initiative used the ISO 14001 Environmental Management framework to structure Arcadia Corp's sustainability efforts."}
]
```

****Example Output:****

```
[
  {
    "entity_id": "EN1",
    "entity_name": "Arcadia Corp",
    "entity_type": "Organization",
    "aliases": ["Arcadia Corp"],
    "description": "Arcadia Corp announced and executed the Greenfield Initiative in March 2023 in response to environmental regulations from the City of Millbrook. Under the Greenfield Initiative, Arcadia Corp achieved an 18 percent reduction in carbon emissions within six months."
  },
  {
    "entity_id": "EN2",
    "entity_name": "Greenfield Initiative",
    "entity_type": "Concept",
    "aliases": ["Greenfield Initiative", "the initiative"],
    "description": "The Greenfield Initiative is a sustainability program launched by Arcadia Corp in March 2023, led by Sarah Chen. The Greenfield Initiative was structured around the ISO 14001 Environmental Management framework and resulted in an 18 percent reduction in carbon emissions for Arcadia Corp."
  },
  {
    "entity_id": "EN3",
    "entity_name": "Sarah Chen",
    "entity_type": "Person",
    "aliases": ["Sarah Chen"],
    "description": "Sarah Chen served as VP of Sustainability at Arcadia Corp and led the Greenfield Initiative, which achieved an 18 percent carbon emission reduction."
  },
  {
    "entity_id": "EN4",
    "entity_name": "City of Millbrook",
    "entity_type": "Location",
    "aliases": ["City of Millbrook", "Millbrook"],
    "description": "The City of Millbrook imposed new environmental regulations in January 2023 that prompted Arcadia Corp to launch the Greenfield Initiative."
  }
]
```

```

    "entity_id": "EN5",
    "entity_name": "ISO 14001 Environmental Management",
    "entity_type": "Method",
    "aliases": ["ISO 14001 Environmental Management", "ISO 14001"],
    "description": "ISO 14001 Environmental Management is the framework used to structure the Greenfield Initiative's sustainability efforts at Arcadia Corp ."
  }
]

```

Note how:

- "the initiative" is captured as an alias for Greenfield Initiative, not as a separate entity.
- "18 percent" and "six months" are NOT extracted -- they are bare metrics/temporal expressions, not entities.
- Descriptions are concise but preserve key facts, relationships, and context.

---Input QA Pairs---

{qa_pairs_json}

---Output Format---

Return a JSON list:

```

[
  {
    "entity_id": "...",
    "entity_name": "...",
    "entity_type": "...",
    "aliases": ["...", "..."],
    "description": "..."
  },
  ...
]

```

Event-extraction prompt (Step 3)

---Role---

You are a Narrative Event Specialist responsible for extracting events and the causal, temporal, and consequential relationships between them from a set of document-grounded Question-Answer pairs.

---Context---

The QA pairs below were generated from a single chunk of a source document using 5W1H-guided analysis. Each QA pair is self-contained with pronouns resolved to explicit entity names.

You are also provided with a list of entities already extracted from these same QA pairs. ****You MUST use the canonical entity names from this list when referring to participants/entities in event descriptions.**** This ensures consistent linking between entities and events downstream.

---Event Definition & Boundary Rules---

An event is a discrete action, occurrence, state change, or happening that can be placed on a timeline.

****Inclusion criteria**** -- extract it if it has at least one of:

- A clear agent performing an action.
- An observable state change (before -> after).
- A temporal anchor (a point or interval in time).

****Exclusion criteria**** -- do NOT extract:

- Static descriptions or attributes of entities (e.g., "Company X is headquartered in Berlin").
- Definitions, classifications, or taxonomic statements.
- Ongoing background conditions with no identifiable start or change point.

****Litmus test:**** Can you place it on a timeline? If not, it is not an event.

---Granularity Guidance---

Extract events at the granularity present in the QA pairs.

Do not summarize multiple distinct events into one mega-event, and do not split a single described action into sub-steps unless the QA pairs treat them

as separate occurrences.

---Cross-QA Synthesis & Deduplication Rules---

A single event may be referenced across multiple QA pairs (e.g., one covers *who*, another *why*). You must synthesize these into a unified event entry.

- ****Merge condition:**** Two QA references describe the SAME event if they share the same core action, the same participants, and the same approximate time/place. Merge them into one entry. Use the most specific event_name and union all details into the description.
- ****Split condition:**** Two references describe DIFFERENT events if the action, outcome, or temporal position differs -- even if participants overlap. Keep them as separate entries.
- ****Never create duplicate events.**** If in doubt, merge.

---Event Type Definitions---

Classify each event using exactly one of the following types. Use "Other" only if none fit.

- **Action:** A deliberate, purposeful act by an agent that does not fit a more specific type below. Use as a fallback only when no other event type applies.
- **Transaction:** An exchange of resources, value, or ownership between parties -- something changes hands (e.g., purchases, acquisitions, grants, donations).
- **Communication:** A deliberate transmission of information to an audience (e.g., announcements, publications, testimonies, press releases). Emphasis is on *sharing* information, not *finding* it.
- **Movement:** A physical relocation of people, forces, goods, or materials from one place to another (e.g., deployments, evacuations, migrations, shipments).
- **Creation:** The bringing into existence of something new that did not previously exist (e.g., founding, building, developing, inventing). If the thing already existed but was unknown, use Discovery instead.
- **Destruction:** The permanent elimination, dismantling, or dissolution of something (e.g., demolitions, decommissionings, dissolutions). If the thing still exists in altered form, use Change instead.
- **Decision:** A deliberate choice, ruling, or judgment that selects a course of action from alternatives (e.g., court rulings, votes, approvals, strategic choices). Emphasis is on the *act of choosing*, not the subsequent execution.
- **Discovery:** The finding or first identification of something that already existed but was previously unknown (e.g., scientific discoveries, detection of threats, uncovering evidence). If the thing was *made new*, use Creation instead.
- **Conflict:** A confrontation, dispute, or antagonistic interaction between opposing parties (e.g., wars, lawsuits, protests, strikes, sanctions).
- **Agreement:** A convergence of parties on shared terms or mutual commitments (e.g., treaties, partnerships, settlements, cease-fires). If the emphasis is on the *exchange of value* rather than *reaching terms*, use Transaction instead.
- **Change:** A transformation or alteration to something already existing (e.g., restructurings, reforms, promotions, state transitions). If something *new* comes into being, use Creation; if something is *permanently eliminated*, use Destruction.
- **Observation:** A passive recording, measurement, or witnessing of a phenomenon without causing it (e.g., surveys, audits, inspections, measurements). If the observed thing is *new or previously unknown*, use Discovery instead.

---Relationship Type Definitions---

Use exactly one of the following relationship types. Choose the most precise fit.

- **`causes`:** A is a necessary or sufficient condition for B. Without A, B would not occur.
- **`triggers`:** A is the proximate catalyst that initiates B at a specific moment. B might have occurred eventually, but A determined the timing.
- **`leads\_to`:** A contributes to B but is not solely responsible. Other factors also play a role.

- ``enables``: A creates the conditions that make B possible, but does not initiate B.
- ``prevents``: A stops B from occurring or continuing.
- ``follows``: B occurs after A with an implied but not explicitly stated causal link. Use when temporal sequence is clear but causality is uncertain.
- ``in_response_to``: B is a deliberate reaction to A by an agent.
- ``escalates``: B is an intensification or expansion of the same dynamic as A.
- ``resolves``: B brings the situation created by A to a conclusion.

****Direction convention:**** Relationships always point forward in narrative time -- ``source_event`` is the earlier or causal event, ``target_event`` is the later or consequent event.

---Instructions---

****Step 1: Event Extraction****

Read through ALL the QA pairs and extract clearly defined, meaningful events that meet the inclusion criteria above. Apply the exclusion rules strictly.

For each event, provide:

- ``event_id``: A short stable identifier (EV1, EV2, EV3, ...).
- ``event_name``: A concise, title-case name.
- ``event_type``: One of the defined event types above.
- ``description``: A comprehensive description synthesized from all relevant QA pairs. ****Name all participants using exact canonical names from the entity list.**** Include what happened, who was involved, when, where, why, and how -- as available. Write in the third person using entity names, never pronouns.
- ``participant_entities``: A list of canonical entity names involved. Every name **MUST** match an ``entity_name`` from the provided entity list exactly.
- ``evidence_strength``: "direct" if the event is explicitly stated in a QA pair; "inferred" if synthesized from indirect references across multiple pairs.

****Step 2: Event Relationship Extraction****

Identify causal, temporal, or consequential relationships between the events extracted in Step 1.

- Only extract relationships that reflect genuine narrative causality or temporal sequence -- not mere co-occurrence or topical similarity.
- The **why** and **how** QA pairs often make these links explicit; leverage them.

For each relationship, provide:

- ``source_event``: The ``event_id`` of the earlier or causal event.
- ``target_event``: The ``event_id`` of the later or consequent event.
- ``relationship_type``: One of the defined relationship types above.
- ``description``: A concise explanation of the causal or temporal link. Use entity names, never pronouns.

****Step 3: Objectivity & Language****

- Write in the third person. Use explicit entity names, never pronouns.
- Output must be in {language}. Retain proper nouns in their original language.

---Pre-Output Validation Checklist---

Before returning your output, verify all of the following:

- Every name in ``participant_entities`` appears verbatim in the provided entity list.
- Every ``source_event`` and ``target_event`` matches an ``event_id`` from your extracted events.
- No two event entries describe the same occurrence (deduplication check).
- No event entry is actually a static fact, attribute, or definition.
- Every relationship reflects a genuine causal or temporal link, not mere topical similarity.
- Every ``event_type`` is one of the defined values above.
- Every ``relationship_type`` is one of the defined values above.

- All descriptions use entity names, not pronouns.

---Worked Example---

****Example Entity List:****

```
[
  {"entity_id": "EN1", "entity_name": "Arcadia Corp", "entity_type": "Organization", "aliases": ["Arcadia Corp"]},
  {"entity_id": "EN2", "entity_name": "Greenfield Initiative", "entity_type": "Concept", "aliases": ["Greenfield Initiative", "the initiative"]},
  {"entity_id": "EN3", "entity_name": "Sarah Chen", "entity_type": "Person", "aliases": ["Sarah Chen"]},
  {"entity_id": "EN4", "entity_name": "City of Millbrook", "entity_type": "Location", "aliases": ["City of Millbrook", "Millbrook"]},
  {"entity_id": "EN5", "entity_name": "ISO 14001 Environmental Management", "entity_type": "Method", "aliases": ["ISO 14001 Environmental Management", "ISO 14001"]}
]
```

****Example QA Pairs:****

```
[
  {"question": "What did Arcadia Corp announce in March 2023?", "answer": "Arcadia Corp announced the launch of the Greenfield Initiative in March 2023."},
  {"question": "Who led the Greenfield Initiative?", "answer": "Sarah Chen, VP of Sustainability at Arcadia Corp, led the Greenfield Initiative."},
  {"question": "Why did Arcadia Corp launch the Greenfield Initiative?", "answer": "Arcadia Corp launched the Greenfield Initiative in response to new environmental regulations imposed by the City of Millbrook in January 2023."},
  {"question": "What happened after the Greenfield Initiative launched?", "answer": "After the Greenfield Initiative launched, Arcadia Corp reduced carbon emissions by 18 percent within six months."},
  {"question": "What framework did the Greenfield Initiative use?", "answer": "The Greenfield Initiative used the ISO 14001 Environmental Management framework to structure Arcadia Corp's sustainability efforts."}
]
```

****Example Output:****

```
{
  "events": [
    {
      "event_id": "EV1",
      "event_name": "Millbrook Environmental Regulations Imposed",
      "event_type": "Change",
      "description": "The City of Millbrook imposed new environmental regulations in January 2023. These regulations prompted Arcadia Corp to take sustainability action.",
      "participant_entities": ["EN4"],
      "evidence_strength": "direct"
    },
    {
      "event_id": "EV2",
      "event_name": "Greenfield Initiative Launch",
      "event_type": "Creation",
      "description": "Arcadia Corp announced and launched the Greenfield Initiative in March 2023 in response to environmental regulations imposed by the City of Millbrook. Sarah Chen, VP of Sustainability at Arcadia Corp, led the Greenfield Initiative. The Greenfield Initiative adopted the ISO 14001 Environmental Management framework to structure Arcadia Corp's sustainability efforts.",
      "participant_entities": ["EN1", "EN3", "EN2", "EN5"],
      "evidence_strength": "direct"
    },
    {
      "event_id": "EV3",
      "event_name": "Arcadia Corp Emission Reduction",
      "event_type": "Change",
      "description": "Arcadia Corp reduced carbon emissions by 18 percent within six months following the launch of the Greenfield Initiative."
    }
  ]
}
```

```

    "participant_entities": ["EN1", "EN2"],
    "evidence_strength": "direct"
  }
],
"event_relationships": [
  {
    "source_event": "EV1",
    "target_event": "EV2",
    "relationship_type": "triggers",
    "description": "The environmental regulations
imposed by the City of Millbrook were the proximate
catalyst for Arcadia Corp launching the Greenfield
Initiative."
  },
  {
    "source_event": "EV2",
    "target_event": "EV3",
    "relationship_type": "leads_to",
    "description": "The launch of the Greenfield
Initiative contributed to Arcadia Corp achieving an
18 percent reduction in carbon emissions."
  }
]
}

```

Note how:

- QA pairs about "who launched it", "why it was launched", and "what framework it used" were merged into a single event (EV2), not split into separate entries -- they describe the same occurrence from different angles.
- EV1 (regulations) and EV3 (emission reduction) are kept separate from EV2 because they have different actions and temporal positions.
- Every name in `participant\_entities` matches an `entity\_id` from the entity list exactly.
- Relationships use `event\_id` values, not event names.
- All descriptions use entity names, never pronouns.

---Extracted Entities (use these canonical names)---
{entities_json}

---Input QA Pairs---
{qa_pairs_json}

---Output Format---
Return a JSON object with exactly this structure:

```

{
  "events": [
    {
      "event_id": "...",
      "event_name": "...",
      "event_type": "...",
      "description": "...",
      "participant_entities": ["ENX", "ENY", ...],
      "evidence_strength": "direct | inferred"
    }
  ],
  "event_relationships": [
    {
      "source_event": "...",
      "target_event": "...",
      "relationship_type": "...",
      "description": "..."
    }
  ]
}

```

E.3 Answer and judge prompts

The shared answer prompt (adapted from LightrAG's default, with output-format constraints and the References section removed and external knowledge forbidden) and the judge prompt. The independent Qwen3-32B and DeepSeek-V3.2 re-scorings (Table 5) used this judge prompt unchanged, through AWS Bedrock (qwen.qwen3-32b-v1:0, deepseek.v3.2) at temperature 0 with reasoning modes off; every one of

the 8,300 calls to each judge returned valid JSON, and the original scores were never overwritten.

Response (answer) prompt

---Role---
You are an AI assistant that answers questions using ONLY the information provided in the Context below. The context is the output of a knowledge-graph / retrieval pipeline run over a book --- it may contain entities, events, relationships, chunk excerpts, or other structured notes extracted from the source text.

---Goal---
Generate an accurate answer to the user query, grounded strictly in the provided Context.

---Instructions---

- Use ONLY facts explicitly stated in the Context. Do NOT invent, assume, or introduce external knowledge.
- Your own knowledge may only be used to form fluent sentences and connect ideas, never to add new information.
- If the Context does not contain enough information to answer, reply exactly with: "{fail_response}"
- Keep the response concise and focused on the query.

---Context---
{context}

---Query---
{question}

Scoring prompt (system)

You are tasked with evaluating a predicted answer against a reference answer for a given question.

Scoring Rubric (1-10 scale):
 10 - Matches ground truth exactly or with a faithful paraphrase; all key points present.
 7 - Mostly correct; minor omissions or wording differences that do not change meaning.
 5 - Partially correct; captures some key points but has major missing information or inaccuracies.
 3 - Mostly incorrect; only small or superficial overlap with the ground truth.
 1 - Off-topic, completely wrong, or hallucinated content.

Output - respond ONLY with valid JSON, no markdown fences:
{ "score": <integer 1-10>, "reason": "<short rationale>" }

Scoring prompt (user)

Question: {question}
 Predicted Answer: {predicted_answer}
 Ground Truth: {ground_truth}

F Worked Example: Retrieved Context

For “Who actually gave Hagrid the dragon egg in the Hog’s Head pub, and why was it significant?” (reference: *Professor Quirrell, disguised, gave Hagrid the egg; significant because Quirrell used Hagrid’s eagerness for a dragon to learn how to get past Fluffy*), we reproduce what each graph method retrieved and answered, so the subgraphs of §5 can be read against the text the answer model received. eSoKG’s context is chunk-free structure

only; E²RAG's and LightRAG's contexts additionally include verbatim source chunks and unmerged (E²RAG) or name-merged (LightRAG) entities.

F.1 eSoKG (scored 9)

Judge rationale: *“correctly identifies Quirrell...and correctly explains the significance...goes beyond the ground truth but doesn't contradict it.”*

eSoKG — retrieved context (excerpt)

This excerpt shows five entities and five events in full, with the remainder listed by name, and the first three relationships in full, with the rest as edges only. The retrieved context passed to the model contained every item in full.

Entities (5 of 63 shown in full)

E3 Hagrid (Person): Rubeus Hagrid is the Keeper of Keys, Grounds, and gamekeeper at Hogwarts, a half-giant expelled in third year but kept on by Dumbledore. He owns Fang and Fluffy (lent to Dumbledore), secretly hatched dragon Norbert, gave Harry a flute used to put Fluffy to sleep, and inadvertently revealed Fluffy's weakness to Quirrell.

E6 The Cloaked Stranger (Person): The cloaked stranger is an unidentified individual who met Hagrid at the Hog's Head, kept his hood up so his face was never seen, and deliberately bought Hagrid drinks while steering conversation toward Fluffy. The stranger offered Hagrid a dragon egg in a card game after extracting the critical information that music could put Fluffy to sleep.

E7 Hog's Head (Location): The Hog's Head is a pub in the village near Hogwarts, described as a place that attracts 'funny folk.' It is where Hagrid met the cloaked stranger, played cards, won Norbert the dragon, and was manipulated into revealing Fluffy's weakness.

E18 Professor Quirrell (Person): Professor Quirinus Quirrell is the stammering, turban-wearing Defense Against the Dark Arts teacher at Hogwarts who became a servant of Voldemort. He let the Halloween troll into Hogwarts, cursed Harry's broom, and attempted to steal the Philosopher's Stone.

E19 Fluffy (Creature): Fluffy is Hagrid's giant three-headed dog, lent to Dumbledore to guard the trapdoor leading to the Sorcerer's Stone. Fluffy can be soothed to sleep by music, a weakness Hagrid accidentally revealed to Quirrell at the Hog's Head in exchange for a dragon egg.

Remaining entities (name only): E1 Hogwarts Library, E2 Hogwarts, E4 Dragon Breeding for Pleasure and Profit, E5 Norbert, E8 Harry Potter, E9 Voldemort, E10 Professor Snape, E11 Ron Weasley, E12 Hermione Granger, E13 Hagrid's Hut, E14 Warlocks' Convention of 1709, E15 Muggles, E16 Hedwig, E17 Herbology, E20 Hogwarts Hospital Wing, E21 Great Hall, E22 The Apothecary, E23 Eeylops Owl Emporium, E24 James Potter, E25 Photo Album, E26 Hagrid, E27 Sorcerer's Stone, E28 Dumbledore, E29 Chocolate Frogs, E30 Trapdoor, E31 Quidditch, E32 Albus Dumbledore, E33 Neville Longbottom, E34 George Weasley, E35 Lee Jordan, E36 Madam Pomfrey, E37 Transfiguration, E38 Nicolas Flamel, E39 Dragon Species of Great Britain and Ireland, E40 From Egg to Inferno A Dragon Keeper's Guide, E41 Draco Malfoy, E42 Gryffindor, E43 Dungeons, E44 Crabbe, E45 Potions, E46 Goyle, E47 Slytherin, E48 Professor Flitwick, E49 Professor McGonagall, E50 Knut, E51 Galleon, E52 Beetle Eyes, E53 Unicorn Horn, E54 Ollivanders, E55 Mr. Ollivander, E56 Ministry of Magic, E57 Mirror of Erised, E58 History of Magic, E59 Anti-Cheating

Spell, E60 Forgetfulness Potion, E61 1637 Werewolf Code of Conduct, E62 Elfric the Eager, E63 Professor Binns.

Events (5 of 30 shown in full)

V4 Hagrid Meets Cloaked Stranger at Hog's Head (Communication; participants E2, E3, E6, E7): Hagrid met The Cloaked Stranger at the Hog's Head. The Cloaked Stranger kept his hood up throughout the encounter, so Hagrid never saw his face. He asked Hagrid about his work, learning that Hagrid's greatest desire was to own a dragon, kept buying Hagrid drinks to impair his judgment, and offered to play cards for a dragon egg on the condition that Hagrid could prove he could handle the dragon.

V1 Hagrid Wins Dragon Egg in Card Game (Transaction; participants E1, E2, E3, E4, E5, E6, E7): Hagrid won a huge black Norwegian Ridgeback dragon egg in a card game with the cloaked stranger at the Hog's Head, who had first established that Hagrid could handle a dragon. The egg later hatched into the dragon Norbert.

V11 Hagrid Reveals Fluffy's Weakness to Cloaked Stranger (Communication; participants E2, E3, E6, E7, E30, E19, E18, E27): During the card game at the Hog's Head, Hagrid accidentally revealed to Professor Quirrell, disguised as a cloaked stranger, that Fluffy could be put to sleep by playing music. Quirrell had kept buying Hagrid drinks to extract this information, offering a dragon egg in exchange.

V2 Harry Potter Realizes Hagrid's Dragon Egg Was Suspicious (Discovery; participants E2, E3, E6, E8, E9, E10, E11, E12): While by the lake during exams, Harry realized the encounter at the Hog's Head was a deliberate setup: a stranger had conveniently offered a dragon egg to Hagrid, who wanted a dragon above all else. Harry concluded the stranger had targeted Hagrid to extract information about the Stone's defenses.

V10 Harry, Ron, and Hermione Visit Hagrid (Communication; participants E2, E3, E6, E7, E19, E5, E8, E11, E12): The trio visited Hagrid and questioned him about what the cloaked stranger looked like on the night he won Norbert. Hagrid revealed that he had told the stranger Fluffy could be put to sleep with music, then looked horrified on realizing he had let the weakness slip again.

Remaining events (name only): V3 Dragon Egg Discovered in Hagrid's Fire, V5 Norwegian Ridgeback Egg Hatching, V6 Hagrid Visits Harry in the Hospital Wing, V7 Hedwig Delivers Hagrid's Note to Harry, V8 Hagrid Buys Harry a Snowy Owl, V9 Hagrid Gives Harry the Photo Album, V12 Trio Relax by Lake After Exams, V13 Hagrid's Suspicious Visit to Library Dragons Section, V14 Hagrid Instructs Trio to Visit His Hut, V15 Hermione Creates Study Schedules, V16 Malfoy Spies on Dragon Hatching, V17 Norbert Grows Rapidly After Hatching, V18 Malfoy Overhears Hatching Conversation, V19 Trio Rushes to Hagrid's Hut, V20 Harry Recovers in Hospital Wing, V21 Harry Shouts Voldemort's Name at Hagrid, V22 First Potions Lesson, V23 Harry and Ron Navigate to Great Hall, V24 Harry and Hagrid Visit The Apothecary, V25 Harry and Hagrid Arrive at Ollivanders, V26 Hagrid Collects Photographs, V27 Trio Attempt to See Dumbledore, V28 Dumbledore Receives Urgent Owl, V29 Harry Retrieves Stone from Mirror of Erised, V30 Year-End Examinations.

Event Relationships (first 3 shown in full)

V4 (Hagrid Meets Cloaked Stranger) --enables--> V11 (Hagrid Reveals Fluffy's Weakness): The stranger's strategy of buying Hagrid drinks and drawing him into conversation created the conditions under which Hagrid revealed Fluffy's weakness.

V11 (Hagrid Reveals Fluffy's Weakness) --follows--> V1 (Hagrid Wins Dragon Egg in Card Game): After the stranger extracted the information about Fluffy, the

card game concluded with Hagrid winning the dragon egg.

V1 (Hagrid Wins Dragon Egg in Card Game) --triggers--> V2 (Harry Realizes Egg Was Suspicious): Harry's reflection that Hagrid had won the egg from a suspicious stranger triggered his realization that the encounter was a deliberate setup.

Remaining relationships (edges only):

V2 --causes--> V10
V10 --causes--> V27
V11 --leads_to--> V6
V11 --leads_to--> V28
V13 --enables--> V3
V14 --leads_to--> V3
V3 --leads_to--> V5
V5 --triggers--> V16
V29 --leads_to--> V20

eSoKG — generated answer

The Cloaked Stranger Who Gave Hagrid the Dragon Egg

Based on the context, the cloaked stranger who gave Hagrid the dragon egg in the Hog's Head was actually **Professor Quirrell**, Voldemort's servant, though he was disguised and kept his hood up throughout the encounter so Hagrid never saw his face.

How It Happened

The encounter was a **deliberate and calculated setup**. The cloaked stranger:

- Kept buying Hagrid drinks to impair his judgment
- Steered the conversation toward Hagrid's work at Hogwarts
- Learned that Hagrid's greatest desire was to own a dragon
- Offered to play cards for a dragon egg, on the condition that Hagrid could prove he could handle a dragon

Why It Was Significant

The dragon egg was essentially a **bribe to extract critical information**. In exchange for the egg, the stranger successfully manipulated Hagrid into revealing that **Fluffy** -- the three-headed dog guarding the trapdoor protecting the Sorcerer's Stone -- could be put to sleep with music.

Harry Potter later realized the entire encounter was a deliberate setup, concluding that the stranger had specifically targeted Hagrid because he "desperately wanted a dragon above all else," making him vulnerable to manipulation. This information ultimately enabled Quirrell to bypass Fluffy and attempt to steal the Sorcerer's Stone.

Hagrid later felt **deeply guilty**, acknowledging he should never have revealed Fluffy's weakness.

F.2 E²RAG (scored 6)

Judge rationale: *“correctly identifies the significance... however, it fails to correctly identify the person as Professor Quirrell (disguised), instead suggesting ‘either Snape or Voldemort’...”* The unmerged context surfaces the dragon’s aftermath rather than the pub exchange.

E2RAG — generated answer

Who Gave Hagrid the Dragon Egg

Based on the context provided, the identity of the person who gave Hagrid the dragon egg was never definitively revealed to Hagrid himself -- the stranger **kept his cloak on and his hood up the entire time**, so Hagrid never saw his face. As Hagrid explained: **"he wouldn't take his cloak off"** and **"I never saw his face, he kept his hood up."**

However, Harry suspected -- and the context strongly implies -- that this was **either Snape or Voldemort in disguise**. Harry concluded: **"it was either Snape or Voldemort under that cloak."**

Why It Was Significant

The dragon egg was significant for several reasons:

- It was used as a manipulation tool.** The cloaked stranger deliberately exploited Hagrid's greatest wish -- owning a dragon -- to gain his trust. The stranger bought Hagrid drinks, got him drunk, and used the egg as a lure in a card game.
- It extracted critical information.** In exchange for the egg (and under the influence of alcohol), Hagrid unwittingly revealed that **playing music would put Fluffy to sleep** -- the one piece of information the stranger needed to get past the three-headed dog guarding the Sorcerer's Stone.
- It was illegal.** Possessing dragon eggs violated wizard law, making the stranger's convenient possession of one deeply suspicious.

As Hagrid himself later admitted with great remorse: **"All for a dragon egg! I'll never drink again!"**

F.3 LightRAG (scored 10)

Judge rationale: *“correctly identifies Professor Quirrell... accurately explains the significance... all key points present.”* Its name-merged Fluffy description explicitly names Quirrell, which the answer draws on.

LightRAG — generated answer

Based on the context provided, the mysterious person who gave Hagrid the dragon egg was a **cloaked stranger** who kept his hood up throughout their meeting at the Hog's Head pub. The stranger played cards with Hagrid, plying him with drinks, and Hagrid won the Norwegian Ridgeback egg in the card game.

The significance of this encounter was profound and deliberately orchestrated. The cloaked stranger -- **later identified as Professor Quirrell** (acting under Voldemort's influence) -- used the dragon egg as a lure to extract critical information from Hagrid. During their conversation, while keeping Hagrid drunk, Quirrell manipulated Hagrid into revealing the secret of **how to get past Fluffy** -- namely, that playing music would lull the three-headed dog to sleep.

As the context states, the dragon egg was "the item the cloaked stranger used to lure Hagrid into a card game, ultimately resulting in Hagrid revealing the secret of how to calm Fluffy." Harry also grew suspicious because a stranger "conveniently had [a dragon egg] in his pocket," recognizing it as too convenient to be coincidental.

This inadvertent disclosure by Hagrid was devastating, as it compromised the security protecting the Sorcerer's Stone and allowed Quirrell to bypass Fluffy. Hagrid later felt **great guilt** for his role in enabling Quirrell's attempt to steal the Stone.