QO-BENCH: Diagnosing Query-Operator-Preserving Retrieval over Typed Event Tuples

Anonymous ACL submission

Abstract

Many real-world questions over business, legal, and scientific corpora are natural-language versions of database-style queries over records latent in text. Existing retrieval-augmented generation (RAG) systems are optimized primarily for semantic relevance, but retrieving plausible passages does not guarantee correct query execution. We introduce QO-BENCH, a diagnostic benchmark for *query-operator question answering* over typed event tuples. The benchmark covers 22,984 news articles and 614 corporate events across 18 query templates, evaluated on 785 questions. Each gold answer is deterministically computed from typed event tuples and scored by recall, with answers matched to the gold tuples by exact match rather than an LLM judge. This design enables operator-level diagnosis such as joins and intersection. We evaluate RAG, ReAct RAG, GraphRAG, and information-extraction-to-SQL under matched conditions, with a long-context oracle ceiling to isolate retrieval failure. A two-axis framework—index-time preservation versus query-time execution—predicts where each paradigm fails, and the results bear it out: systems retrieve relevant text but discard the typed values operators need, and the deployable paradigm ranking inverts across operators, with similarity retrieval leading on filter/project and extraction-to-SQL on intersection and counting. Even given the gold evidence, a long-context oracle stays far from saturated, so operator execution—not retrieval alone—is a core bottleneck that a stronger answer model does not remove. QO-BENCH reframes the goal from passage relevance to query-operator-preserving retrieval.

1 Introduction

Many high-value information needs in business, law, and policy are expressed in natural language but behave like database queries. A user may ask, *Which firms were buyers in M&A announcements in both 2018 and 2022?* or *Which firms had a CEO change within 30 days of an M&A announcement?* The records these questions range over are not given in a table; they lie *latent across a text corpus*, reported piecemeal across many articles. Because no single article pre-aggregates them for an arbitrary query, such questions cannot rely on one summarizing passage. They require selecting events that satisfy constraints, assigning roles, anchoring dates, joining evidence across conditions and other aggregations. We call this setting *query-operator question answering* (QO-QA): natural-language questions specify database-style operators over records. Relational databases support such questions through typed attributes, query plans, and set semantics (Codd, 1970, 1972; Gray et al., 1997). However, many real corpora are not clean databases. They are news articles, filings, contracts, reports, emails, or analyst notes, where the relevant records must first be recovered from text. QO-QA therefore lies at the boundary between semantic parsing and retrieval-augmented generation: the question demands database-style execution, but the evidence resides in unstructured documents.

Retrieval-augmented generation (RAG) (Lewis et al., 2020; Karpukhin et al., 2020) is typically built around a different contract: retrieve passages that are semantically relevant to a question, then ask a language model to synthesize an answer. This contract is powerful for passage lookup, but QO-QA exposes a structural mismatch. For *Which firms were buyers in M&A announcements in both 2018 and 2022?*, a system must identify M&A events, normalize firm names, distinguish buyers from targets, attach announcement dates, construct buyer sets for each year, and intersect them. Top-k retrieval may return plausible passages, but semantic relevance alone does not guarantee role correctness, temporal correctness, set completeness, or count correctness.

The capability needed here is *operator-preserving retrieval*: retrieval that preserves the typed values required to execute the operators expressed in the question. This framing separates two sources of failure. On the corpus side, an index may fail to preserve operator-relevant attributes. On the query side, a system may fail to convert the natural-language question into an executable plan involving the operators it induces. Standard RAG and ReAct-style RAG (Yao et al., 2023) leave most operator execution to the generator. GraphRAG (Edge et al., 2024) introduces entity-relation structure and summaries, but summaries may compress away necessary attributes. Information-extraction-to-SQL systems (Li et al., 2021; Yu et al., 2018; Scholak et al., 2021) execute operators explicitly, but only after committing to a schema and extraction pipeline whose coverage and normalization may be incomplete. Thus, QO-QA is not simply "text-to-SQL over news" or "RAG with more documents": it asks whether retrieval architectures can bridge natural-language operator intent and latent event structure in text.

We introduce QO-BENCH, a diagnostic benchmark for QO-QA over typed event tuples. The benchmark spans 18 query templates evaluated on a stratified sample of 785 questions. The questions are instantiated over 22,984 news articles covering 614 corporate events. [1] Each gold answer is deterministically computed from event tuples with event type, entity, role, anchor date, counterparty, and provenance fields. This gives QO-BENCH an explicit denotational gold standard. Systems return structured final answers—entity lists, event lists, ordered lists, counts, grouped results—which are canonicalized and compared with the gold denotation by exact match—not an LLM judge—and scored with template-specific recall. This design lets us attribute each failure to a specific operator rather than to answer phrasing. Our experiments compare RAG, ReAct RAG, GraphRAG local and global search, and information-extraction-to-SQL. Our goal is not to crown a winner, but to produce a failure profile: which operators each paradigm supports, which it approximates poorly, and which it cannot reliably execute.

Our experiments yield a consistent diagnosis. No paradigm dominates: the deployable paradigms' ranking *inverts* across operators, with similarity retrieval (RAG, ReAct RAG) strongest on filter/project and IE→SQL strongest on intersection and counting. The failures also separate along the framework's two axes—similarity retrieval loses *coverage*, GraphRAG preserves structure but not exact *values*, and IE→SQL covers events yet cannot *execute* cross-event joins beyond its schema. And the ceiling is itself operator-bound: even fed the gold evidence, a long-context oracle stays far from saturated (∼4% on set intersection), and a stronger or more heavily reasoning answer model does not lift it—operator execution, not retrieval alone, is a core bottleneck.

Contributions. First, we formulate QO-QA, where questions specify database-style operators over records latent in text, and identify operator preservation as a central retrieval property. Second, we introduce QO-BENCH, a benchmark with deterministic typed event-tuple gold answers and role-, date-, and counterparty-aware denotations over provenance-attested events. Third, we use the QO-QA framework to decompose representative paradigms along its two axes— index-time preservation and query-time execution—and, against a long-context oracle ceiling that separates retrieval from answer-synthesis failures, locate where each paradigm fails.

2 Related Work

QO-BENCH builds on work in retrieval-augmented, multi-document, temporal, list, and aggregation question answering, but targets a different diagnostic unit. Dense, late-interaction, and retrieval-augmented models make large-corpus retrieval central to QA (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Lewis et al., 2020; Izacard and Grave, 2021), while multi-document and long-context benchmarks test whether models combine evidence across passages (Shaham et al., 2023; Bai et al., 2024). Multi-hop and agentic retrieval methods further decompose questions into iterative retrieval steps (Yang et al., 2018; Trivedi et al., 2022; Ho et al., 2020; Yao et al., 2023; Press et al., 2023; Trivedi et al., 2023). These settings are related, but they usually evaluate whether systems find and synthesize relevant evidence, rather than whether retrieval preserves the typed values needed to execute database-style operators.

Several benchmarks cover parts of this problem.

[1] Corporate events provide a useful substrate because they naturally exhibit the operator surface we want to test. The same QO-QA structure appears beyond business news, including legal case aggregation, biomedical evidence synthesis, and policy monitoring

Benchmark	Typed event-tuple gold	Deterministic gold (no judge)	Per-operator diagnosis	Matched archs. + oracle	Cross-doc completeness
AGGBench	✗ (entity)	✓	Aggregate only	Partial (no oracle)	✓
MEBench	✗ (entity)	✓	Reasoning types	✗	Partial
FanOutQA	✗	✗ (LLM judge)	✗	✗	Partial
TLQA	✗ (lists)	Partial	Temporal / list	✗	Partial
ChronoQA	✗	✗ (LLM/human)	Temporal types	Partial (RAG)	Partial
LIQUID / ListQA	✗ (lists)	Partial	✗	✗	✗
RAGBench	✗	✗ (TRACe)	✗	✗	✗
FinanceBench	✗	Partial	✗	✗	✗
QO-BENCH	✓	✓	✓ (6 operators)	✓ (5 paradigms + LC)	✓

Table 1: Comparison with related benchmarks. QO-BENCH is, to our knowledge, the first benchmark to combine deterministic typed event-tuple gold (no LLM judge), per-operator failure attribution, and a matched multi-architecture comparison against an oracle ceiling.

FanOutQA evaluates fan-out multi-document questions (Zhu et al., 2024); MEBench studies cross-document reasoning (Lin et al., 2025); TLQA evaluates time-referenced list construction (Dumitru et al., 2025); ChronoQA studies temporal-sensitive RAG with absolute, aggregate, and relative temporal questions (Chen et al., 2025); and RAGBench evaluates general RAG behavior and attribution (Friel et al., 2024). List-QA datasets such as LIQUID evaluate questions with multiple non-contiguous answers (Lee et al., 2023), while financial QA benchmarks such as FinQA, TAT-QA, FinanceBench, and FAITH emphasize numerical reasoning, tabular reasoning, evidence-grounded answers, or tabular faithfulness over financial documents (Chen et al., 2021; Zhu et al., 2021; Islam et al., 2023; Zhang et al., 2025). AGGBench is closest because it studies aggregation over unstructured text as a completeness-oriented find-all task (Zhu et al., 2026). Unlike AGGBench's entity-level aggregation setting, QO-BENCH evaluates deterministic query execution over typed event tuples.

QO-BENCH is also related to structure-based retrieval and information extraction. GraphRAG organizes extracted entities and relations into graphs (Edge et al., 2024), which can improve enumeration and global sense-making but may lose exact operator-relevant values such as dates, roles, and counts. Event extraction, semantic parsing, and text-to-SQL make operators executable by converting text into structured records and questions into programs (Wang et al., 2020; Li et al., 2021; Yu et al., 2018; Scholak et al., 2021), but depend on predefined schemas and complete normalized tuples. We therefore compare these paradigms under matched conditions (§5). The goal is not to claim that aggregation, temporal reasoning, list answers, or financial QA are new, but to test whether retrieval architectures can preserve and execute the typed event values required by natural-language query operators. Table 1 summarizes this distinction.

3 A Framework for Query-Operator QA

3.1 QO-QA as Denotational Retrieval

We formalize QO-QA as a derivation from corpus to answer. A corpus entails events with attributes; a question selects events, projects fields, and applies an operator; and a system is judged by whether it recovers the denotation required by the question.

Let $C = \{d_1, \ldots, d_n\}$ be a document corpus and let $E = \{e_1, \ldots, e_m\}$ be the latent event set entailed by C. Each event $e \in E$ has an attribute set $A(e)$ whose elements $a = (n, v)$ pair an attribute name with a value derivable from the corpus. Attribute names are open-ended: for an M&A event, for example, relevant names may include acquirer, target, seller, announcement date, completion date, deal value, jurisdiction, and regulatory status.

A question q is a typed query specification

$$q = (k, \mathcal{D}_q, \phi, \pi, \alpha),$$

where $k \geq 1$ is the arity and $\mathcal{D}_q \subseteq E_{\tau_1} \times \cdots \times E_{\tau_k}$ is the typed tuple domain induced by the event types in the question. For $k = 1$, we identify the tuple (e) with the event e when this causes no ambiguity. The predicate ϕ selects tuples from $\mathcal{D}_q$ using attribute constraints and, for $k \geq 2$, tuple-level conditions such as shared entities, temporal windows, or announce–complete deal identity. The projection π returns structured records with component labels, and α is an operator such as list, count, group.

The selected tuple relation and projected relation are

$$R_q = \{\vec{e} \in \mathcal{D}_q : \phi(A(\vec{e})) = 1\}, \quad (1)$$

$$P_q = \langle \pi(A(\vec{e})) : \vec{e} \in R_q \rangle, \quad (2)$$

where $A(\vec{e}) = (A(e_1), \ldots, A(e_k))$ and $\langle\cdot\rangle$ denotes a multiset that preserves tuple identity and multiplicity, as required by counting, grouping, and ordering.

The gold answer is the denotation

$$y^*(q) = \mathsf{Ans}_E(q) = \alpha(P_q). \tag{3}$$

Depending on α, $y^*(q)$ may be a list or set of entities or events, an ordered list, a matched event-pair set, a count, a grouped table; when $R_q = \emptyset$, the empty list, set, or table, or a count of 0.

A document set $D \subseteq C$ is sufficient for q if it attests the attributes needed to evaluate ϕ and the attributes returned by π for the selected tuples. For tuple queries, this includes fields needed to evaluate joins, such as normalized entity IDs, roles, anchor dates, counterparties, and deal-linking identifiers. Thus, QO-QA stresses set-complete retrieval: a system must surface the selected tuples in R_q and the fields needed to project and aggregate them. In our benchmark, many multi-document questions are multi-event and multi-article retrieval problems; in broader settings, a single event may also require evidence composed across multiple documents.

3.2 Preservation and Execution

QO-QA imposes two architectural requirements. First, the index must preserve operator-relevant values. Second, the runtime must execute the operator structure expressed by the question.

Index-time preservation. The relevant issue is not whether the original text is stored, but whether operator-relevant values are recoverable through the system's query-time interface in typed, executable form. Let $\iota(C)$ denote the index a system builds over C—its stored, queryable representation of the corpus. We call an index $\mathcal{S}$-preserving for an attribute set $\mathcal{S}$ if, for every event e and every attribute $a \in A(e)$ whose name lies in $\mathcal{S}$, the value of a can be recovered from $\iota(C)$ by a well-defined query-time procedure under the system's *retrieval budget*—the bounded evidence (e.g. top-k chunks or a fixed number of retrieval hops) it may surface per query (Appendix F). Full open-schema preservation, where $\mathcal{S}$ contains all possible attribute names in $A(e)$, is an idealized target.

Query-time execution. At query time, the system must parse the question into an operator specification, retrieve tuples and fields conditioned on that specification, and apply α over the projected relation. Parsing errors produce the wrong plan; retrieval errors omit tuples or fields; aggregation errors occur when evidence is present but the system still miscounts, misgroups, misorders, or fails to intersect sets. For large selected relations, set-complete retrieval is often the bottleneck: once relevant tuples are outside the retrieved context, downstream generation cannot reconstruct them from available evidence.

We apply these two requirements as diagnostic axes for five representative paradigms: RAG, ReAct RAG, GraphRAG in local and global query modes, and IE→SQL, which builds on event extraction and text-to-SQL. The selection is not exhaustive, but it spans distinct points along both axes. Table 2 summarizes the decomposition.

RAG and ReAct RAG preserve article text, embeddings, and chunk-level metadata, but leave role, stage, counterparty, and aggregation operators to the generator. GraphRAG preserves entity- and relation-level structure in a summary-driven graph, but exact operator-relevant values such as dates, stages, roles, and counts may be compressed into prose. IE→SQL materializes typed attributes and executes operators explicitly, but only for fields covered by its frozen schema $\mathcal{S}$. The unoccupied design point is corpus-side coverage with query-time typed execution beyond a fixed schema; QO-BENCH diagnoses which parts of this pipeline current paradigms support and which remain missing.

3.3 A Tractable Subclass for Evaluation

The latent formulation above allows open-ended queries over open-ended attribute sets, which is difficult to evaluate deterministically. We therefore evaluate a controlled subclass.

(1) Schema-bounded queries. Let $\tau(e)$ denote the event type of event e. For each event type t, we fix a finite schema $S(t)$ containing the fields used by our templates. Every template restricts ϕ and π to fields in the schemas of the component event types.

(2) Single-document-attestable events. An event $e \in E$ is admitted to the operational set $\widehat{E} \subseteq E$ iff some article attests every schema field in $S(\tau(e))$ for e, after canonicalization and conflict resolution.

These restrictions make the benchmark easier than the fully open-ended setting: the relevant schema is known in advance, and each event is individually recoverable from a single article. Failures on this restricted substrate therefore provide conservative evidence of architectural difficulty. We

	Indexing		Querying		
Paradigm	**Method**	**Preservation scope**	**Parse**	**Retrieve**	**Aggregate**
RAG	Text chunks + embeddings	Text + chunk metadata	No typed parse	Semantic	Generator
ReAct RAG	Text chunks + embeddings	Text + chunk metadata	Sub-question plan	Iterated semantic	Generator
GraphRAG-local	Entity graph + summaries	Entity/relation summaries	Entity match	Neighborhood	Generator
GraphRAG-global	Entity graph + summaries	Entity/relation summaries	No typed parse	Community map	Fixed reduce
IE→SQL	Schema-typed tuples	$A(e) \cap \mathcal{S}$	NL → SQL	SQL selection	SQL operators

Table 2: Paradigm decomposition along the two requirements of §3.2. *Indexing* (**Method**, **Preservation scope**) shows how the index is built and which part of $A(e)$ it preserves in queryable form; *Querying* (**Parse**, **Retrieve**, **Aggregate**) shows how the runtime dispatches the three query-time subtasks.

do not claim a formal dominance relation between recall on this subclass and recall in the fully open-ended setting, because the two settings may differ in query distribution, event distribution, and retrieval budget.

4 The QO-BENCH Benchmark

4.1 Corpus, Events, and Ground Truth

Instantiating the tractable subclass of §3.3, QO-BENCH pairs a financial-news corpus with structured corporate-event ground truth. The corpus is the NASDAQ subset of FNSPID (Dong et al., 2024). Event candidates come from S&P Capital IQ Key Developments, a structured corporate-event stream recording event type, firms and roles, and dates, taken over 2010–2023. We align each S&P event to FNSPID articles by two filters—the event's firm ticker and an event-type–specific date window around its anchor date (Appendix B)—producing a set of candidate articles per event. A 3-of-3 LLM judge then attests, per event, which candidates genuinely describe it (§4.2); an event enters the operational set $\widehat{E}$ only when an article attests its schema fields. The resulting benchmark contains 22,984 FNSPID articles[2] and $|\widehat{E}| = 614$ single-article-attestable events across eight types: M&A announcement, completion, cancellation, and rumor; CEO change; CFO change; IPO; and stock split. Figure 1 summarizes the construction pipeline.

Event-type definitions are anchored in public records, including SEC Form 8-K Items, Securities Act §5, and NYSE/NASDAQ listing rules, and are supplied verbatim to every paradigm (Appendix D). Ground truth is represented as typed event tuples keyed by public identifiers rather than as text spans. Table 3 gives the common tuple fields. The queryable schema $S(t)$ for event type t contains the fields used by templates. Provenance is retained as evidence metadata but is not itself a query predicate. This representation enables deterministic denotational evaluation while avoiding span-match ambiguity. Licensing details are in Appendix A.

Field	Description
Firm	Normalized focal firm identifier, e.g., ticker, CIK, or canonical name.
Event type	One of eight event categories; M&A lifecycle stages are separate types.
Date	Anchor date used for filtering and ordering.
Role	Buyer, target, seller, or executive role, depending on event type.
Counterparty	Optional normalized counterparty.
Provenance	Supporting article IDs; evidence metadata, not a queryable field in $S(t)$.

Table 3: Common event-tuple fields.

Deterministic evaluation. For each question, the gold answer is the denotation $y^*(q) = \alpha(P_q)$ over the operational event set $\widehat{E}$, using the notation of §3.1. Evaluation therefore compares the system's returned answer $f(q, C)$ (its answer for q on corpus C) directly with $y^*(q)$, without an LLM-as-a-judge at scoring time: matching to the gold denotation is by exact match on canonicalized values, and answers are scored by recall. The template-specific recall metric is defined in §5.

[2]The reported 22,984 counts distinct FNSPID articles; because one article can match several events, event–article links are more numerous (on average 1.13 events per article).

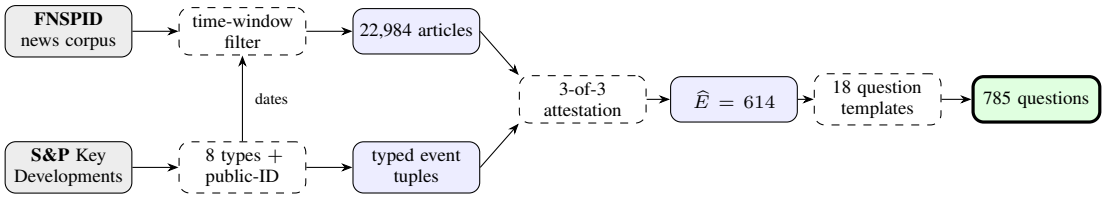

Figure 1: QO-BENCH construction pipeline. S&P Capital IQ events time-window-filter the FNSPID (Dong et al., 2024) corpus; 3-of-3 judge attestation aligns the two into the operational event set $\widehat{E}$ (614 single-article-attestable events), over which 18 templates instantiate the 785-question benchmark with deterministic gold denotations.

4.2 Validating the Judge Consensus

The operational set $\widehat{E}$ is constructed using unanimous 3-LLM attestation: an event–article pair is accepted only when all three judges agree that the article attests the event's schema fields. To validate this high-precision proxy, three expert annotators re-examined a stratified sample of 221 accepted event–article pairs (up to 30 per event type), each pair receiving two independent labels. Counting an accepted pair as correct only when both annotators confirm it, the 3-of-3 consensus reaches 94.1% precision, confirming that admitted pairs are reliable. Because the pool contains only accepted pairs, this is a precision-only check (recall is not estimable). Per-type precision, inter-annotator agreement, and the disagreement analysis are reported in Appendix C.

4.3 Template Taxonomy

QO-BENCH contains 18 primary templates in two capability classes. Primary evaluation uses a stratified sample capped at 50 questions per template, totaling 785 questions: 200 from four Capability A templates and 585 from fourteen Capability B templates. **Capability A (filtered retrieval)** questions require selection and projection under typed predicates. Examples include listing a firm's M&A announcements within a time window or returning firms with IPOs in a year. These templates test type-matched event recall, entity disambiguation, and role-aware retrieval.

Capability B (compositional operations) questions add an operator over filtered events. Examples include finding firms with both a CEO change and an M&A announcement within 30 days, identifying M&A buyers in two different years, selecting the earliest IPO in a quarter, counting firms above an event-frequency threshold, grouping events by quarter, and returning type-labeled unions. These templates test temporal joins, intersections, ordering, counting, grouping, and multi-type aggregation. Appendix E lists every template with its signature, sample size, example, and diagnostic target.

5 Experiments and Analysis

We compare five deployable paradigms against a retrieval-free **long-context oracle** (**LC-oracle**) ceiling that feeds each question the 3-of-3 attested documents supporting its gold answer. Holding the answer LLM (Qwen3.6-27B) fixed, the gap from each deployable paradigm to this ceiling isolates its retrieval contribution.

5.1 Setup

Templates are scored by recall with a ±7-day date tolerance (absorbing news-vs-anchor date drift). Recall is computed on the *covered subset*: each gold denotation is restricted to events attested in the corpus ($\widehat{E}$), so a system is not penalized for gold events whose evidence the corpus lacks. This recall is our main metric. For leakage control, no paradigm sees question templates at index time and all use fixed decoding; in particular IE→SQL's schema is generated from event definitions alone, frozen, and used verbatim with no manual corrections (full configurations in Appendix F).

5.2 Main Results

Overall, the LC-oracle ceiling reaches 52.2%; among deployable paradigms IE→SQL is strongest at 37.9%, ahead of RAG (25.2%) and ReAct RAG (23.9%), with GraphRAG global search (3.8%) and local search (0.9%) near the no-context floor (0.6%; Table 11). Table 4 reports recall by operator family and capability; per-template recall is deferred to Appendix G (Table 9). The headline is not the ranking but its *operator-dependence*: no paradigm dominates across families, and the deployable ranking inverts from one operator to the next.

Three patterns organize the results. **(1) Even the ceiling is operator-bound.** With gold evidence in

Operator family	LC-oracle	RAG	ReAct RAG	GR-local	GR-global	IE→SQL
Capability A						
Filter / project	77.6	53.6	55.4	3.5	7.3	50.6
Role / type	35.3	18.2	51.1	0.0	6.3	55.6
Cap A (all)	67.0	44.8	54.3	2.6	7.1	51.9
Capability B						
Temporal join	52.3	11.1	15.7	0.3	0.0	21.1
Ordering	59.9	29.8	27.8	0.5	5.8	17.8
Intersection	3.9	6.0	9.5	0.4	2.2	50.9
Count / group	56.2	30.0	3.4	0.3	5.1	51.2
Cap B (all)	47.1	18.5	13.5	0.3	2.7	33.1
Overall	52.2	25.2	23.9	0.9	3.8	37.9

Table 4: Per-operator-family and capability recall (%), ±7-day tolerant, question-weighted (micro).

context, LC-oracle is far from saturated (52.2%): 77.6% on filtering but 3.9% on intersection, so operator *execution*, not only retrieval, is a bottleneck. **(2) Filtering is broadly solved.** On filter/project, RAG, ReAct RAG, and IE→SQL each recover about half the ceiling (53.6, 55.4, 50.6; GR-local is the exception, near zero). **(3) The ranking inverts on composition.** IE→SQL dominates intersection (50.9 vs. RAG/ReAct 6.0/9.5) and counting/grouping (51.2 vs. ReAct 3.4): SQL executes these natively, while a generator must reconstruct them from prose—on intersection IE→SQL even *exceeds* the LC-oracle, whose LLM cannot reliably intersect from gold chunks—the oracle bounds retrieval and generation, not operator execution, so an explicit SQL executor can legitimately surpass it.

Re-scoring the LC-oracle ceiling with two further answer models—from open-weight to frontier scale and light to heavy reasoning—leaves the hardest operators just as hard (Table 11): intersection stays near 4% for all three. A stronger or more heavily reasoning answer model does not lift the ceiling: operator execution is ill-suited to free-form generation—an architectural mismatch, not a capacity limitation.

5.3 Localizing Failure: Preservation and Execution

The framework's two axes—index-time *preservation* and query-time *execution*—imply that paradigms fail in different places.

Preservation fails in two ways. The first is *coverage*: gold-article retrieval recall (Table 10) measures whether the supporting evidence is surfaced at all, and RAG and ReAct RAG reach just 32.4% and 38.4% of gold articles, with Cap B coverage (∼27–30%) far below Cap A: for similarity retrieval the corpus-side axis is lost *before any operator runs*. The cause is not only relevance ranking but the fixed top-k cutoff itself: it hard-caps the candidate set, so once a query's gold answer spans more distinct events than top-k surfaces, set-completeness is unattainable by construction—which is why coverage collapses on the many-event Cap B aggregations.

The second is *value-fidelity*: surfacing the evidence is not enough if the index does not also keep the exact values an operator consumes. GraphRAG fails here. Its index preserves entity–relation *structure* and community summaries, but the LLM-generated reports compress away the precise dates, roles, and counts: GR-local scores ≤3.5% on every value-dependent family despite organizing the whole corpus—organization is not preservation. GR-global makes the point sharply: its map-reduce forwards essentially the entire corpus (article-level coverage ≈100% by construction; Table 10), yet still reaches only 3.8% overall, losing the same values in summarization rather than in retrieval. With the corpus-side index held fixed, the local-to-global gap (0.9 → 3.8%) isolates the query side: global's fixed map-reduce surfaces more community reports than local's entity-match neighborhood, lifting most families a few points (e.g. filtering 3.5 → 7.3); only cross-event temporal joins stay near 0% in both modes, since the values that would link events were never preserved. GraphRAG thus fails the *value-fidelity* facet of preservation, distinct from the coverage facet that bounds RAG.

Execution is the query-side axis: even the LC-oracle, fed the gold documents directly, scores only 3.9% on intersection—here execution, not retrieval, is the limiting factor. IE→SQL covers 64.8% of

gold documents (near-uniform across Cap A/B), so its residual losses lie not on coverage but on this same axis, on cross-event temporal joins: it scores 0% on the announce-to-complete cross-stage join (B.1.4), where linking an announcement to its completion needs a cross-stage linkage identity the extractor populates on 93% of M&A announcement records but only $\sim 4\%$ of completion records, capping announce-to-complete pairing at $\sim 4\%$, and equally 0% on the cross-type CEO-change/M&A join (B.1.2), whose join key it likewise never materializes. Decomposing IE→SQL bounds the ceiling: extraction recall tops out at 75.8% (event level, above end-to-end SQL recall), and 50% of residual misses are the extractor declining to emit a record from a thin in-article mention—paradigm-honest conservatism, not SQL error. This is the structural ceiling of schema-bounded extraction, and it sits on the query axis, not on coverage.

The three deployable families thus fail at three distinct points—similarity retrieval on *coverage*, GraphRAG on *value-fidelity*, and schemaful execution on the *query* axis—the first two both facets of index-time preservation, exactly as the framework predicts.

5.4 Implications

The results bear out the operator-specific failure the framework implies: no paradigm both preserves operator-relevant values across the corpus and executes typed operators over them. Similarity retrieval optimizes relevance over completeness, summary-based graphs discard the exact roles, stages, dates, and counts they index, and schema-bound extraction caps what can be queried. The design target is therefore not larger context or better generation, but broad corpus-side coverage combined with query-time typed execution beyond a fixed schema.

This diagnosis suggests several architectural directions. One is to treat plan-and-execute retrieval as an IR primitive: a query planner would infer the operator structure of the question, retrieve candidate events for coverage, materialize the fields needed for a query-specific schema, and execute aggregation over the resulting temporary relation. A second direction is schema-with-residual coverage, where a fixed event schema handles common operators while residual text retrieval captures attributes outside the schema. A third is late-binding entity and event resolution, triggered only when the query requires cross-event joins or intersections. Finally, retrievers should be trained not only for relevance but also for set completeness.

More broadly, QO-QA isolates capabilities that single-document reading comprehension and open-domain QA do not stress: exact filtering, attribute preservation, cross-document composition, and aggregation.

6 Conclusion

This paper argues that query-operator QA is not naive QA with more documents: a natural-language question specifies a computation over latent event records. To study this setting, we introduced QO-BENCH, a diagnostic benchmark with 18 typed query templates over 22,984 financial-news articles and 785 questions.

We also proposed a two-axis framework that separates index-time preservation from query-time execution. In matched experiments, RAG, ReAct RAG, GraphRAG, and IE→SQL exhibit various operator-specific failures. The LC-oracle gap shows that many failures arise before generation, when retrieval fails to expose the records over which the answer must be computed.

QO-BENCH reframes cross-document QA as *schemaful information retrieval with natural-language input and output*. We hope it supports future work on operator-preserving retrieval systems that combine broad corpus coverage with query-time typed execution beyond fixed schemas.

Limitations

QO-BENCH uses corporate events as operational ground truth. The tested operators—filtering, joining, intersection, counting, grouping, and ordering—are domain-general, but absolute results may not transfer to domains with weaker public records, less standardized event definitions, or noisier entity resolution. The benchmark also uses template-generated questions, which provide controlled operator probes but underrepresent the linguistic diversity of natural user queries. Although answers are computed deterministically from normalized event tuples, the tuples may still contain errors from source ambiguity, date conventions, event-stage boundaries, or entity normalization; we mitigate these risks through public identifiers, provenance, unanimous attestation, and human validation. Finally, each paradigm admits many implementations and scores depend on the answer LLM, so our results should be interpreted as matched-

condition architecture diagnostics rather than exhaustive or model-invariant performance estimates.

Ethics and Responsible Benchmarking

QO-BENCH is built from public corporate news and public corporate-event records. It does not use non-public personal data; any executive names or firm affiliations come from public disclosures or news reports. Corporate news coverage is uneven and tends to overrepresent large publicly listed firms, English-language markets, and media-salient events. This skew should be considered when interpreting coverage or event-frequency statistics.

The benchmark is intended for retrieval-architecture evaluation, not for investment, legal, compliance, or employment decisions. System outputs should not be treated as verified financial facts without human review and source checking. To respect article-text licensing, we release article identifiers, metadata, public-identifier event tuples, questions, prompts, and evaluation scripts; redistribution of full article text remains governed by the originating publishers' licenses.

References

Yushi Bai and 1 others. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *ACL*.

Zhiyu Chen and 1 others. 2021. FinQA: A dataset of numerical reasoning over financial data. In *EMNLP*.

Ziyang Chen and 1 others. 2025. A question answering dataset for temporal-sensitive retrieval-augmented generation. *Scientific Data*.

Edgar F. Codd. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Edgar F. Codd. 1972. Relational completeness of data base sublanguages. In Randall Rustin, editor, *Data Base Systems: Courant Computer Science Symposia Series 6*, pages 65–98. Prentice-Hall.

Zihan Dong, Xinyu Fan, and Zhiyuan Peng. 2024. FNSPID: A comprehensive financial news dataset in time series. In *KDD*.

Alexandru Dumitru, Venktesh V, Adam Jatowt, and Avishek Anand. 2025. Evaluating list construction and temporal understanding capabilities of large language models. In *ICTIR*.

Darren Edge and 1 others. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. RAGBench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. 1997. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53.

Xanh Ho and 1 others. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*.

Pranab Islam and 1 others. 2023. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*.

Vladimir Karpukhin and 1 others. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. LIQUID: A framework for list question answering dataset generation. In *AAAI*.

Patrick Lewis and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Qian Li and 1 others. 2021. A survey on deep learning event extraction: Approaches and applications. *arXiv preprint arXiv:2107.02126*.

Teng Lin and 1 others. 2025. MEBench: Benchmarking large language models for cross-document multi-entity question answering. In *EMNLP*.

Ofir Press and 1 others. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*.

Torsten Scholak and 1 others. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *EMNLP*.

Uri Shaham and 1 others. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of EMNLP*.

Harsh Trivedi and 1 others. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*.

Harsh Trivedi and 1 others. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *ACL*.

Xiaozhi Wang and 1 others. 2020. MAVEN: A massive general domain event detection dataset. In *EMNLP*.

Zhilin Yang and 1 others. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Shunyu Yao and 1 others. 2023. ReAct: Synergizing reasoning and acting in language models. In *ICLR*.

Tao Yu and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *EMNLP*.

Mengao Zhang, Jiayu Fu, Tanya Warrier, Yuwen Wang, Tianhui Tan, and Ke-wei Huang. 2025. FAITH: A framework for assessing intrinsic tabular hallucinations in finance. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF)*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *ACL*.

Fengbin Zhu and 1 others. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *ACL*.

Haojia Zhu and 1 others. 2026. Aggregation queries over unstructured text: Benchmark and agentic method. *arXiv preprint arXiv:2602.01355*.

A Data Sources, Licensing, and Public-Identifier Release

QO-BENCH draws news articles from FNSPID (Dong et al., 2024), a public dataset of NASDAQ financial news, and corporate events from S&P Capital IQ Key Developments, a proprietary structured-event feed accessed under license.

Disclaimer. The S&P Capital IQ Key Developments feed is proprietary and may not be redistributed. The released benchmark therefore contains no S&P content: neither raw event records, vendor record identifiers (e.g., `keyDevId`, internal firm IDs), nor S&P headlines or situation summaries. Each ground-truth event is published instead as a *public-identifier tuple* — the participating firm keyed by a public identifier (stock ticker, SEC CIK, or LEI), the event type drawn from our eight public-record-anchored definitions, the anchor date, role, and counterparty — together with provenance article IDs into FNSPID. These tuples are recoverable from public records and carry no proprietary text, so the benchmark is vendor-independent and freely redistributable while remaining a faithful denotational gold standard. Reconstructing the original vendor records requires a separate S&P Capital IQ license.

Event type	Days before	Days after
M&A announcement	7	14
M&A completion	7	90
M&A cancellation	7	30
M&A rumor	14	14
CEO change	14	60
CFO change	14	60
IPO	7	60
Stock split	7	14

Table 5: Per-event-type date windows around the S&P anchor date used to collect candidate FNSPID articles, in days before/after the anchor. Windows are asymmetric because news typically trails the legal/announcement date.

B Corpus Construction

FNSPID subset. We use the NASDAQ subset of FNSPID (Dong et al., 2024): 2,953,850 URL-deduplicated articles pre-tagged with 8,553 distinct stock symbols. We apply no explicit date filter; although FNSPID is nominally dated 1999–2023, its NASDAQ coverage before 2010 is sparse, so in practice the corpus spans 2010–2023—matching the period of the S&P events. Date selection within this span happens per event below.

Article matching. Each S&P event is matched to candidate FNSPID articles by exact ticker (FNSPID articles carry a stock symbol; no alias or fuzzy normalization is applied), restricted to a per-event-type asymmetric date window around the anchor date (Table 5). Articles with empty bodies are dropped, and duplicates are removed by URL, keeping the occurrence closest to the anchor. The anchor is the legal/announcement date for each type; because news typically trails legal completion by several weeks, windows are widened after the anchor rather than symmetric.

Attestation funnel. Three judges—Gemma-4-31B-IT, Qwen3.6-27B, and gpt-oss-120B—independently label each (event, article) pair, and a pair is attested only on unanimous 3-of-3 confirmation. From the 16,414 S&P events across the eight types in 2010–2023, we select a 1,376-event subset to bound judging cost, producing 25,888 candidate (event, article) pairs; 1,591 pairs (6.1%) are attested 3-of-3, yielding $|\widehat{E}| = 614$ distinct single-article-attestable events.

C Judge-Consensus Validation

The operational set $\widehat{E}$ relies on a unanimous 3-of-3 LLM-judge consensus. To validate it, three expert

annotators (using the same event definitions as the judges) labeled a stratified sample of 221 accepted (event, article) pairs (up to 30 per type); every pair received two independent labels.

Precision. Because the pool contains only accepted pairs, the check is precision-only (recall is not estimable without a separately drawn negative pool). We count an accepted pair as correct only when both of its annotators confirm it—any single rejection, including a split vote, counts as incorrect—a conservative criterion. Under it the 3-of-3 consensus reaches 94.1% precision ($208/221$). Precision is at least 90% on seven of the eight event types; only M&A completions are lower, at 73.3%, where an article often references the closing only with vague timing.

Inter-annotator agreement. Agreement is computed on the 209 pairs co-labeled by one annotator pair (the remaining 12, co-labeled by a different pair, are unanimously positive and leave κ undefined). On these pairs raw agreement is 96.2% and Cohen's $\kappa = 0.538$. The moderate κ despite high raw agreement is a prevalence effect: the accepted-only pool is dominated by positive labels, which inflates chance agreement and deflates κ; it does not indicate low reliability.

Disagreement. Boundary errors concentrate on whether an article truly *attests* the event rather than merely *mentions* it: (i) M&A completions referenced only with vague timing inside earnings-call transcripts or analyst recaps; (ii) IPO articles dated before the offering, using forward-looking language ("plans to raise," "set to price"); and (iii) tangential mentions of CEO/CFO transitions in earnings-call acknowledgements. These cases affect boundary decisions in $\widehat{E}$; once an event tuple is admitted, gold answers are computed deterministically from normalized tuple fields.

D Event Definitions

Each event type is defined by reference to a public-record disclosure class (Table 6), keeping the ontology vendor-independent and identical across paradigms. This document is supplied verbatim to every paradigm as shared task supervision; full operational scope and edge cases accompany the released code.

Type	Record	Date	Role
M&A announce	8-K 1.01	announce	buyer/target
M&A complete	8-K 2.01	close	buyer/target
M&A cancel	8-K 1.02	terminate	buyer/target
M&A rumor	NYSE §202.03	publish	target
CEO change	8-K 5.02	report	—
CFO change	8-K 5.02	report	—
IPO	Sec. Act §5	report	—
Stock split	NYSE §703.02	effective	—

Table 6: The eight event types, each anchored to a public-record disclosure class (SEC 8-K items, the Securities Act, or exchange listing rules). The anchor date is the canonical date per type; *role* applies to M&A (rumor records only the target) and is null otherwise.

E Template Catalog

Tables 7 and 8 list all 18 templates with their signatures, output types, per-template sample sizes, an example question, and the operator each diagnoses.

F Paradigm Configurations

All deployable paradigms share the corpus, the event-definitions document, and the same answer LLM—Qwen3.6-27B (vLLM, reasoning mode enabled), decoding held fixed; only the retrieval architecture differs.

RAG retrieves with a hybrid stack—dense (Qwen3-Embedding-4B) plus BM25, then a reranker (Qwen3-Reranker-4B)—feeding the top-30 chunks to the answer LLM. **ReAct RAG** uses the LangChain `create_react_agent` recipe with light prompt adaptation: the agent issues up to five retrieval calls over the same hybrid stack, accumulating evidence across rounds before answering. **GraphRAG** (Edge et al., 2024) extracts entities and relations, clusters them into Leiden communities, and produces LLM-generated community reports, queried in two modes—*local* (entity match $\rightarrow$ neighborhood $\rightarrow$ related-community fetch) and *global* (fixed map-reduce over community reports). We run the `graphrag` package with unmodified source and generous (not minimal) index settings; the only prompt change, shared with all paradigms, prepends the event-definitions document. Local and global share the corpus-side index but differ on the query side, isolating the query-side contribution.

IE$\rightarrow$SQL proceeds in three stages: (i) the database schema is *generated by an LLM (GPT-5.5, T=0) from the event definitions alone*—never exposed to question templates—and frozen verbatim; a pre-specified escape hatch for documented

ID	Signature	Output	N	Example	Diagnoses
A.1.1	(E, W)	List[Event]	50	List all CEO changes in Q1 2023.	Type-matched event recall in a window.
A.1.2	$(E, W) \rightarrow \pi_{\text{firm}}$	List[Entity]	50	Which firms had IPOs in 2018?	Event-to-firm deduplication.
A.2.1	(E, F, W)	List[Event]	50	List Microsoft's M&A announcements 2020–2023.	Entity disambiguation under filter.
A.3.1	(E, R, W)	List[Entity/Event]	50	Firms that were M&A targets in 2020.	Role-aware retrieval.

Table 7: Capability A (filtered retrieval) templates. N is the per-template count in the stratified evaluation sample (at most 50 per template).

ID	Signature	Output	N	Example	Diagnoses
B.1.1	(E, F, W, dir)	List[Event]	50	M&A announces in 2022 before Microsoft–Activision.	Anchor + relative window.
B.1.2	$([E_1, E_2], W, \Delta)$	List[EventPair]	50	CEO change + M&A announce within 30d, 2022–23.	Symmetric temporal join.
B.1.3	$([E_t, E_f], W, \Delta)$	List[EventPair]	50	CFO change $\rightarrow$ M&A announce within 90d.	Directional lag.
B.1.4	(W, Δ)	List[Deal]	39	M&A deals announced 2022 closing within 180d.	Announce$\rightarrow$complete identity.
B.1.5	$(F, [E_1, E_2], W)$	Int	50	Days between MSFT–ATVI announce and close?	Date arithmetic on self-join.
B.2.1	(E, W, pos)	OrderedList	50	First IPO of Q1 2018?	Extremal selection.
B.2.2	$(F, \{E_1 \ldots\}, W)$	OrderedList	50	Tesla CEO changes, M&A, splits 2015–23.	Multi-type interleaving.
B.3.1	$([E_1, E_2], W)$	List[Entity]	30	Firms with CEO change AND M&A announce in 2023.	Cross-type intersection.
B.3.2	$(E, R, [W_1, W_2])$	List[Entity]	46	M&A buyers in both 2018 and 2022.	Cross-window same-role.
B.3.3	$([E_1, E_2], R, W)$	List[Entity]	15	M&A target in both announce and complete, 2020.	Cross-type same-role.
B.4.1	(E, W, N)	List[Entity]	16	Firms with $\geq$ 3 CEO changes, 2010–20.	Count threshold.
B.4.2	(E, R, W, N)	List[Entity]	39	M&A buyer $\geq$ 4 times, 2015–23.	Role-aware count threshold.
B.4.3	(E, W, b)	List[Event]+b	50	CEO changes 2023 grouped by quarter.	Time bucketing.
B.4.4	$(\{E_1 \ldots\}, W)$	List[Event]+t	50	M&A announces + IPOs Q1 2023, labeled.	Type-label union.

Table 8: Capability B (compositional operations) templates. N is the per-template count in the stratified evaluation sample (at most 50 per template).

manual corrections was not invoked (zero corrections); (ii) a separate LLM (Qwen3.6-27B, T=0, reasoning mode) extracts event tuples from each article into the frozen schema with provenance; (iii) per-template SQL skeletons translate questions to SQL executed against the events database, with no LLM at query time. Stage (i) is the schema-leakage control: IE$\rightarrow$SQL is evaluated as an intrinsic test of schemaful retrieval, not as a hand-tuned upper bound.

LC-oracle bypasses retrieval and feeds each question its gold-supporting chunks linked via provenance. It is a ceiling, not a deployable paradigm.

Full prompts, chunking parameters, embedding model, retrieval k, hop budgets, GraphRAG version and indexing settings, the IE$\rightarrow$SQL schema-generation prompt and raw output, extraction prompts, and SQL skeletons accompany the released code.

G Additional Results

Per-template recall. Table 9 reports per-template recall for all paradigms—the resolution behind the operator-family aggregate of Table 4.

Gold-article retrieval recall. Table 10 measures whether the gold-attesting evidence reaches a paradigm at all, decoupled from whether the operator is then executed: for each question we compute the fraction of its gold-attesting articles whose content the paradigm surfaces into its answering context (for IE$\rightarrow$SQL, whose event the extraction stage recovers), averaged by template. The LC-oracle is 100% by construction. For IE$\rightarrow$SQL, the extraction stage recovers roughly two-thirds of gold articles, well above its end-task recall (Table 9), localizing the dominant loss to the query/composition layer rather than to coverage.

Answer-model robustness and the no-context floor. Table 11 reports the per-operator profile of the LC-oracle ceiling across three answer models (Qwen3.6-27B; DeepSeek v4-flash; v4-pro Think Max) alongside the no-context floor (Qwen, parametric only, zero articles). The cross-model invariance of the ceiling is discussed in §5; beyond it, the floor isolates retrieval's per-family contribution. No-context recovers near-zero recall on every family—retrieval and context account for nearly all of the ceiling, from 77.6% on filtering down to its lower operator-bound limits.

Template	LC-oracle	RAG	ReAct RAG	GR-local	GR-global	IE→SQL
Filter / project						
A.1.1	72.3	46.9	49.6	3.4	3.6	51.8
A.1.2	83.2	43.9	62.5	3.4	15.7	54.0
A.2.1	77.3	70.0	54.0	3.7	2.7	46.0
Role / type						
A.3.1	35.3	18.2	51.1	0.0	6.3	55.6
Temporal join						
B.1.1	70.5	0.0	0.0	1.2	0.2	38.1
B.1.2	41.7	7.0	16.1	0.0	0.0	0.0
B.1.3	44.0	3.0	18.7	0.0	0.0	38.9
B.1.4	22.6	9.1	0.0	0.0	0.0	0.0
B.1.5	76.0	36.0	40.0	0.0	0.0	24.0
Ordering						
B.2.1	54.7	30.3	23.7	0.0	8.3	10.3
B.2.2	65.2	29.3	31.8	1.0	3.3	25.2
Intersection						
B.3.1	7.8	11.1	23.6	0.7	1.7	39.4
B.3.2	0.0	2.9	3.3	0.0	3.3	53.6
B.3.3	7.8	5.6	0.0	1.3	0.0	65.7
Count / group						
B.4.1	28.1	20.8	2.1	0.0	4.2	58.3
B.4.2	43.6	34.6	6.8	0.0	6.4	59.8
B.4.3	68.5	33.2	1.2	0.8	4.1	45.8
B.4.4	62.7	26.2	3.2	0.2	5.3	47.4

Table 9: Per-template recall (%) under ±7-day date tolerance (no provenance); 785-question stratified sample, Qwen3.6-27B answer model.

Paradigm	Cap A	Cap B	Overall
LC-oracle	100.0	100.0	100.0
RAG	47.6	27.2	32.4
ReAct RAG	61.8	30.4	38.4
GR-local		N/A	
GR-global		≈100†	
IE→SQL	62.6	65.6	64.8

Table 10: Gold-article retrieval recall (%) under ±7-day matching: per-question coverage of gold-attesting articles; capability columns are question-weighted (micro) means over GT-applicable templates. For IE→SQL this is extraction-stage coverage; for the retrieval paradigms it is the fraction of gold articles surfaced into the answer context. LC-oracle is 100% by construction (fed gold chunks). GR-local gold-article retrieval recall is not well-defined—its entity-graph neighborhood answers from community summaries that expose no clean mapping back to source articles, so per-article gold coverage cannot be computed (N/A). †GR-global retrieval is non-discriminative—it forwards essentially the entire corpus (all community reports) to the LLM map step, so article-level retrieval recall is ≈100% by construction; its information loss occurs in community-report summarization, not in retrieved-set membership.

	LC-oracle (ceiling)			No-ctx
Operator family	Qwen	v4-flash	v4-pro	(floor)
Filter / project	77.6	78.2	77.3	1.1
Role / type	35.3	35.2	35.4	0.6
Temporal join	52.3	34.9	38.4	0.0
Ordering	59.9	59.0	61.6	2.1
Intersection	3.9	4.4	3.8	0.5
Count / group	56.2	55.8	59.9	0.1
Cap A	67.0	67.5	66.8	1.0
Cap B	47.1	39.8	42.7	0.5
Overall	52.2	46.8	48.8	0.6

Table 11: Answer-model robustness and the no-context floor (%, ±7-day tolerant, question-weighted micro). The LC-oracle ceiling is re-scored with three answer models—Qwen3.6-27B (open-weight, ∼5K reasoning tokens/question), DeepSeek v4-flash (no reasoning), and v4-pro (Think Max, ∼24K reasoning tokens/question). The hardest operators are model-invariant—intersection collapses to ∼4% for all three even with gold articles supplied—so the ceiling is operator-bound, not capacity-bound; models differ mainly on Cap B (temporal joins). The No-ctx column (Qwen, parametric only, zero articles) is the floor: retrieval contributes nearly all recall on every family.