Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

RAG Query Rewriting: Does It Preserve the Question?

> Reliable Context is a kernel of truth: a system of record for organizations that fuse private and public information and must defend every value they act on. One canonical record per real thing, a cited source and an observation date behind every value, unknowns declared rather than left blank, and any past state of the record reproducible.

Published by Reliable Context · 2026-10-20 · rag query rewriting · query rewriting in rag · query transformation · query decomposition · conversational query rewriting · rag evaluation · constraint preservation · query drift · paired evaluation

Original article: https://reliablecontext.com/articles/rag-query-rewriting-preserve-question-constraints



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes: How Rewriting Alters the Question
4. Why Retrieval Relevance Does Not Certify Fidelity
5. Implementation Considerations: An Acceptance Protocol
6. Data Analysis and Evidence
7. Implications and Future Directions
8. Frequently Asked Questions (FAQs)
9. Conclusion

Executive Summary

A RAG (retrieval-augmented generation) query rewriter is usually justified by a retrieval or answer-quality gain, but that gain is measured against the wrong object if the rewritten search has quietly changed the question. The central finding of this review is that the rewriting papers reviewed here optimize retrieval or downstream answer scores, not fidelity to the original request; results on retriever weaknesses are indirect evidence for why constraint loss matters, not measurements of rewriters. Ma et al. (2023) train a rewriter with reward tied to the reader's answer and to whether retrieved text contains the gold answer (aclanthology.org), and CONQRR (Wu et al., 2022) uses a reward that directly optimizes retrieval (aclanthology.org). Its authors state that the rewrites are meant to improve retrieval rather than be used by end users (aclanthology.org). Neither paper certifies that entity, date, polarity, quantity or requested comparison survive. Query drift was already a named hazard in classical expansion work (romisatriawahono.net), and a human evaluation of one LLM rewriter found that its rewrites conveyed the user's real search intent in 84.8% to 89.4% of cases across three datasets (ar5iv.labs.arxiv.org).

The evidence that such qualifiers matter is independent and consistent. On NevIR, no bi-encoder exceeds 12% paired accuracy (arxiv.org); temporal retrieval methods prioritize matching exact dates, which the authors call a shortcut (arxiv.org); and a 2026 benchmark reports dense retrievers satisfying numerical constraints only 16.3% of the time (www.sciopen.com). Expansion can also hurt: Weller et al. report that it helps weaker retrievers but generally harms stronger ones (arxiv.org), and a March 2026 study found rewriting lowered nDCG@10 by 9.0 percent on FiQA while raising it by 5.1 percent on TREC-COVID (arxiv.org).

This article therefore proposes an acceptance protocol, not a measured result. Annotate an intent contract from the original request before any model rewrite is seen; freeze the corpus, models and settings; compare original search, a human standalone reference and the candidate rewrite on identical cases; build paired fixtures that change one decisive qualifier; and score constraint preservation, evidence retrieved for the original question, and supported-answer correctness separately. Report the share of requests preserving every critical constraint beside the average, because an average can hide a decisive date or entity mutation.

Statistics should be paired at the request level. McNemar's test is a standard paired test for binary outcomes (web.engr.oregonstate.edu), matched pairs per topic are the convention in retrieval testing (ciir-publications.cs.umass.edu), and correlated observations within a cluster call for cluster-aware resampling

(www.nber.org). The release rule is three-way: ship the rewriter where preregistered materiality rules hold, fall back to the original search where they do not, and ask for clarification where the referent cannot be resolved. All numbers in the worked examples are hypothetical and labelled as such.

Introduction and Background

Query rewriting in RAG places a transformation between the user's request and the retriever. Weaviate's documentation describes it as reformulating the original user query to make it more suitable for retrieval (weaviate.io). The decision addressed here is narrower than whether rewriting works in general. An enterprise context engineer building an evidence-bearing application must decide whether a measured retrieval gain is acceptable when the rewritten search may have changed the entity, date, polarity, scope or requested comparison. This is an evaluation protocol, not a rewriter implementation tutorial, and every fixture and scoring suggestion below is a proposed method rather than a measured outcome.

Four terms are used throughout, and keeping them apart is the point of the protocol.

- **Question intent** is what the requester asked: referents, time scope, quantities and units, negation, the required comparison, the evidence requested and any ambiguity that is genuinely unresolved. It is fixed from the original request and the permitted conversation.
- **Search expansion** adds terms, hypothetical text or alternative phrasings to raise the chance of retrieving relevant material. It need not be a paraphrase. A hypothetical document used for embedding lookup rather than the raw query is a legitimate search device even though it is not the question (developers.llamaindex.ai).
- **Standalone rewrite** converts a context-dependent turn into a self-contained question. CANARD defines the task as rewriting a context-dependent question into a self-contained question with the same answer (aclanthology.org). The "same answer" clause is a fidelity requirement that retrieval-oriented training objectives do not enforce.
- **Clarification** is a request back to the user when the referent or scope cannot be determined. It is the correct response when a guess would silently redefine the question.

The distinction between expansion and rewrite matters because a strict literal-paraphrase gate would reject useful behavior, while a pure retrieval metric would accept harmful behavior. The protocol resolves this by judging the transformation against an explicit contract, then judging the evidence and the answer against the same contract. Retrieval relevance cannot certify question fidelity: a passage can be highly relevant to a rewritten question about a different year, a parent company instead of a subsidiary, or a decrease instead of an increase, and conventional metrics such as Hit Rate, MRR and NDCG score relevance to whatever query was issued.

This report does not claim that nobody evaluates rewriting. The Milvus guide, a vector database vendor's documentation, covers sub-queries and step-back prompts at a high level (milvus.io), and the academic literature measures task gains carefully. The contribution proposed here is a preregistered question-constraint ledger and paired qualification fixtures that make fidelity failures countable. A reader who needs a retriever,

vector index or RAG framework should select from those categories; none of those replaces the protocol below, which needs a frozen, versioned corpus snapshot so that every test run retrieves from the same evidence base.

Key Changes: How Rewriting Alters the Question

Search expansion changes the query, and sometimes the topic

The classical name for harmful expansion is query drift, defined in the query-expansion literature as the alteration of the focus of a search topic caused by improper expansion (romisatriawahono.net). The same survey warns that the effect may be more serious when the added term is a proper noun (romisatriawahono.net), which is the usual case for entity-bearing enterprise questions. Mitra, Singhal and Buckley reported in 1998 that refining the feedback document set often prevents the query drift caused by blind expansion (people.eng.unimelb.edu.au), which shows that drift was a known, measurable hazard before LLMs.

LLM-based expansion inherits the hazard. Query2doc, published at EMNLP 2023, reports that LLM-generated pseudo-documents boost BM25 by 3% to 15% on ad-hoc IR datasets (aclanthology.org). Weller et al., testing eleven expansion techniques, twelve datasets and twenty-four retrieval models, found that expansion improves scores for weaker models but generally harms stronger models (arxiv.org). A March 2026 study of prompt-only single-step rewriting reports that rewriting degrades nDCG@10 by 9.0 percent on FiQA and improves it by 5.1 percent on TREC-COVID (arxiv.org). The authors attribute the FiQA loss to rewriting that replaces domain-specific terms in queries that already matched well. The practical reading is that the sign of the effect depends on corpus and retriever, so a gain reported elsewhere does not transfer to a given site.

Standalone rewrites in conversation resolve referents, and can pick the wrong one

Conversational rewriting is the stage most likely to introduce a silent entity change, because the rewriter must choose among referents in earlier turns. TopiOCQA reports that 28% of turns require retrieving a document different from the previous turn (ar5iv.labs.arxiv.org), so topic switches, where a wrong carry-over of an entity is most likely, are frequent. The TREC CAsT 2019 organizers note that off-the-shelf coreference models struggled with CAsT topics more than expected (ar5iv.labs.arxiv.org). LLM rewriters narrow the gap with human rewriting but not completely: the LLM4CS human evaluation found that its rewrites successfully convey the user's real search intent in 85.5%, 89.4% and 84.8% of cases across three datasets (ar5iv.labs.arxiv.org). By arithmetic on those figures, roughly one rewrite in nine to one in seven did not; the paper does not state that number. The same authors acknowledge that LLM rewrite generation lacks awareness of the downstream retrieval process (ar5iv.labs.arxiv.org).

Retrieval-optimized rewrites are not paraphrases, by design

The two primary papers discussed here optimize different objectives, and neither is a fidelity guarantee. Ma et al. (EMNLP 2023) propose Rewrite-Retrieve-Read, motivated by the observation that there is inevitably a gap between the input text and the needed knowledge in retrieval (aclanthology.org). The arXiv record carries the title Query Rewriting for Retrieval-Augmented Large Language Models (arxiv.org), while the ACL Anthology uses "in"; the abstract is the same on both pages. The trainable rewriter learns from reader feedback by

reinforcement learning, and for open-domain QA the reward gives credit if the retrieved content hits the answer and penalizes a miss (aclanthology.org). The paper's case study shows what the resulting rewrites look like: queries that put a keyword forward (aclanthology.org). The authors frame the objective as adapting the retrieval query to improve the reader on downstream tasks (aclanthology.org). The paper reports no separate measurement of whether a rewrite preserves the original question. That is a property of the objective, not a defect: the paper makes no fidelity claim to breach. Its reported gains are task gains, and they are not uniform: rewriting improved most settings except the social sciences on MMLU (aclanthology.org).

CONQRR (Wu et al., EMNLP 2022) rewrites conversational questions with a reward function that directly optimizes retrieval using reinforcement learning (aclanthology.org). The paper states that its aim is to improve retrieval performance instead of directly being used by end users (aclanthology.org). Its error analysis is directly relevant: among 50 sampled examples where CONQRR (mix) ranked the gold passage worse than the human rewrite with the dense encoder, 34% involved additional information with a misinterpretation of the user request (aclanthology.org). Ye et al. state the property that matters here: the rewritten query should preserve the meaning of the original query (ar5iv.labs.arxiv.org). No benchmark discussed in this review measures that property together with retrieval gain on enterprise constraints, which is why the protocol separates them.

Decomposition and step-back change the requested comparison

Decomposition splits one request into several searches. The Milvus guide says an LLM can break a complicated query down into simpler sub-queries (milvus.io); LlamaIndex documents a sub-question engine that gathers intermediate responses and synthesizes a final response (docs.llamaindex.ai); Azure AI Search documents LLM-based planning that breaks a complex query into smaller, focused subqueries (learn.microsoft.com); and LangChain's MultiQueryRetriever writes a set of queries with an LLM and returns the unique union of all retrieved documents (reference.langchain.com). A unique union has no step that checks the retrieved set against the original question. The RAG-Fusion authors report in their abstract that some answers strayed off topic when the generated queries had insufficient relevance to the original query (arxiv.org), a qualitative single-study observation.

Decomposition fidelity is itself imperfect. The Break benchmark defines question understanding as the ability to break a question into the requisite steps for computing its answer (transacl.org), and automatic decomposition into such steps is an error-prone task in its own right. Step-back prompting abstracts the question into higher-level concepts and reports gains on MMLU Physics and Chemistry of 7% and 11% (iclr.cc) and on TimeQA by 27% (iclr.cc). Its authors also catalogue an error class, Context Loss, where the model response loses context from the question (arxiv.org). A step-back question about a general principle can drop the specific year or entity that the user asked about, which is a plain failure of the requested scope even when the final answer is fluent.

Why Retrieval Relevance Does Not Certify Fidelity

The paired-qualifier fixtures proposed below rest on work showing that retrievers and similarity metrics are weakest where one decisive qualifier flips the answer set.

Negation, exclusion and set operations

NevIR builds document pairs that differ only by negation and reports that most IR models do not consider negation, performing the same or worse than a random ranking (arxiv.org). Its metric is paired: a pair counts only if the ranking flips correctly for both queries, because models tended to rank one document above the other for both queries (arxiv.org). No bi-encoder scores higher than 12% on it. A rewriter that drops a "not" or turns "excluding" into a keyword hands the retriever a different question, and a retriever that already struggles with negation will not signal the change.

Time scope

TimeQA's authors note that lots of facts can evolve with respect to time (datasets-benchmarks-proceedings.neurips.cc), and its best long-document model reached 46% accuracy against 87% for humans (datasets-benchmarks-proceedings.neurips.cc). A temporal-RAG study that perturbs the temporal relation (before, after, as of) finds that all methods prioritize matching exact dates, indicating a shortcut (arxiv.org). A date constraint can therefore be lost in two places: the rewriter may normalize or drop it, and the retriever may match the literal date string rather than the relation.

Quantities, units and entities

A 2026 benchmark of numerical constraints reports that state-of-the-art dense retrievers achieve only 16.3% accuracy in numerical constraint satisfaction (www.sciopen.com). The authors of that benchmark frame the result as specific to the retrievers they tested, and it should not be read as a statement about every deployed retriever. Entity handling is the other recurrent hazard. On EntityQuestions, a DPR dense retriever scores 49.7% against 72.0% for BM25 on average (arxiv.org). If a rewriter replaces a rare entity string with a descriptive phrase or a commoner alias, the retriever may lose its strongest signal; if it substitutes a parent for a subsidiary, the retriever will find fluent, relevant, wrong evidence.

Similarity metrics do not certify meaning preservation

Semantic-equivalence scores are tempting as automatic fidelity checks. BERTScore computes token similarity with contextual embeddings instead of exact matches (iclr.cc). Work on learned evaluation metrics argues that models of this family are insensitive to negations (arxiv.org). That finding comes from text-generation evaluation and was not measured on rewrite-versus-original pairs. They justify one rule: similarity may triage candidates for review, but cannot certify fidelity.

Implementation Considerations: An Acceptance Protocol

Everything in this section is a proposed method. No benchmark is supplied for it, none of the figures are measured, and the protocol has not been run on any corpus.

Step 1: annotate the intent contract before seeing any rewrite

Two reviewers annotate each original request, plus only the conversation turns the application is permitted to use, without access to any model rewrite. The contract records:

- **Referents**: entities with admissible aliases verified against the frozen sources, and for conversation, which earlier-turn referent is intended or whether it is unresolved.
- **Time scope**: dates, ranges and relations (before, after, as of, during), and the reference date for any "current" language.
- **Quantities and units**: thresholds, comparison operators, currency or measurement units, and aggregation (annual versus cumulative).
- **Polarity**: negation and exclusion, and status qualifiers such as approved versus proposed.
- **Required comparison**: the sets compared and the direction (increase versus decrease).
- **Requested evidence**: the kind of source the requester asked for, if stated.
- **Unresolved ambiguity**: a flag for requests where the correct behavior is clarification, with the admissible interpretations listed.

Disagreements are adjudicated, and agreement before adjudication is reported, as TREC CAsT 2019 did when its manual rewrites were compared and the disagreements adjudicated to a canonical form (ar5iv.labs.arxiv.org). Report raw agreement and a chance-corrected coefficient per constraint type, since most constraints in most requests are uncontentious and chance-corrected agreement behaves differently from raw agreement under imbalance. Each constraint also receives a materiality label (critical or non-critical) now, so that the release rule cannot be tuned to the results. Record every ambiguous label and each admissible alternative rewrite, so the later verdict is "outside the admissible set" and not "different from one reference."

Step 2: freeze the experimental conditions

Freeze the corpus snapshot and source versions, the rewriter, retriever and generator versions, prompts, sampling settings, top-k and the token budget, and fix a reference date for any "current" language. Three arms run on identical cases:

- **Original search**: the request, or the minimal permitted conversation context, goes to the retriever unchanged.
- **Human standalone reference**: a reviewer writes a standalone question from the contract. The reference must not leak answer terms, so the writer works from the request and the contract, not from the evidence. Human rewrites are not guaranteed to be the best search queries, so this arm is a fidelity reference, not a retrieval ceiling.
- **Candidate rewrite**: the production rewriter, condenser or decomposer, with every emitted sub-query logged.

Step 3: build paired fixtures that change one decisive qualifier

Paired fixtures take one request and produce a minimal twin that differs in a single decisive qualifier, in the spirit of NevIR's document pairs and AmbigQA's practice of writing disambiguated questions as minimally edited modifications of the prompt question (ar5iv.labs.arxiv.org). Generic information examples suffice. Suggested pair families:

- **Date**: before versus after a named date, or fiscal versus calendar year.
- **Organization**: parent versus subsidiary, or two entities sharing a name.
- **Direction**: increase versus decrease, or a rise in a rate versus a rise in a count.

- **Status**: approved versus proposed, draft versus final.
- **Aggregation**: annual versus cumulative, per unit versus total.
- **Referent**: one of two conversational referents, with the twin conversation swapping which entity was mentioned last.
- **Unresolved**: a conversation where two referents are equally plausible and clarification is the correct response.

A pair passes only if the rewrite, the retrieved evidence and the answer all differ appropriately between twins, or the system asks for clarification where the contract says it should. Scoring the pair as a unit follows NevIR's rationale for paired accuracy: a system can return the same result for both twins. Include a deliberate expansion control as well: cases where the best rewrite adds vocabulary or a hypothetical answer without altering the contract. Such an expansion should pass if the evidence and answer remain accountable to the original contract. Without this control, an overstrict gate would reject useful behavior and push teams toward no rewriting at all.

Step 4: score the transformation, the evidence and the answer separately

Five scores, never merged into one:

- **Constraint preservation** equals preserved applicable constraints divided by annotated applicable constraints. A constraint is preserved if the rewrite states it, or carries it as a filter outside the query text, in a form the adjudicators judge equivalent to an admissible interpretation. Constraints added that are not in the contract count as alterations.
- **Critical-constraint completeness** is the fraction of requests in which every critical constraint is preserved. It is reported beside the average so a decisive date or entity mutation cannot hide in a high mean.
- **Evidence recall against original intent** is the share of contract-required evidence items retrieved, judged against the original contract, not the rewritten query. This is where retrieval relevance and question fidelity separate. Near-miss evidence is a failure distinct from missing evidence.
- **Supported-answer correctness** requires that the answer is correct under the contract and that its claims are supported by the retrieved evidence. Claim-level attribution checking is a downstream diagnostic and is not repeated here.
- **Clarification correctness**: asking when the contract marks the request unresolved, and not asking otherwise. A system that clarifies everything passes the first test and fails the second, so report both.

For decomposition, record the rewrite-stage view per sub-query and for the union: every sub-query can preserve its constraints while the requested comparison across them is dropped. *Table 1* below shows the proposed original-to-rewrite constraint matrix. All rows are hypothetical.

Request ID	Source turn	Constraint	Allowed interpretations	Rewrite change	Evidence retrieved	Verdict
H-014 (Hypothetical Example)	Turn 1	Time: "after the 2023 policy change"	Dates strictly after the change date stated in the source	"policy change" kept; "after" dropped	Pre- and post-change documents, mixed	Altered, critical
H-027 (Hypothetical Example)	Turn 3	Referent: "its revenue"	Subsidiary named in turn 2, or parent named in turn 1; ambiguous	Rewrite names the parent	Parent revenue only	Unresolved referent guessed; should clarify
H-031 (Hypothetical Example)	Turn 1	Polarity: "items not yet approved"	Exclusion of approved items	"not yet approved" replaced by "under review"	Approved and unapproved items	Altered, critical
H-040 (Hypothetical Example)	Turn 1	Expansion: adds aliases for the same entity	Any alias verified in the frozen sources	Adds two verified aliases, keeps entity and date	Same evidence set plus one additional item	Preserved; useful expansion
H-052 (Hypothetical Example)	Turn 2	Quantity: "above 5 percent, annual"	Annual rate above 5 percent	Rewrite says "5 percent growth"	Cumulative figures	Altered, critical

The matrix is the artifact reviewers reuse. Its columns separate what was asked, what the rewrite changed and what came back, so a reader can see that H-040 is an expansion that passes and H-027 is a clarification case, not a retrieval error.

Step 5: localize where the failure first appears

For each failed request, label the first stage at which the contract is lost: rewrite, retrieval or generation. A rewrite that preserves every constraint but retrieves the wrong evidence is a retrieval failure; a rewrite that drops the date is a rewrite failure even if the retriever happened to return the right documents. Retrieval-quality aspects (context relevance, noise robustness) and generation-quality aspects are separable in principle, and the RGB authors report that models show some noise robustness but struggle with negative rejection, information integration and dealing with false information (ojs.aaai.org). The wider RAG fault-isolation matrix is the right downstream tool once the rewrite stage is cleared, and is not duplicated here.

Step 6: report paired gains and regressions with request-level uncertainty

The unit of analysis is the request, and a request with five correlated constraints is one observation. Cameron, Gelbach and Miller note that cluster-robust methods permit heteroskedasticity and within-cluster error correlation (www.nber.org), and that rejection rates of ten percent under standard methods can be reduced to the nominal five percent by cluster bootstrap procedures (www.nber.org). In IR testing, system scores are paired per topic (ciir-publications.cs.umass.edu). Smucker, Allan and Carterette found little practical difference among the randomization, bootstrap and t tests (ciir-publications.cs.umass.edu), recommended the

randomization test as the preferred test (ciir-publications.cs.umass.edu), and said use of the Wilcoxon and sign tests should be discontinued for measuring a difference between means (ciir-publications.cs.umass.edu). Berg-Kirkpatrick et al. use the paired bootstrap for NLP (aclanthology.org).

For a binary outcome per request (supported-correct or not), McNemar's test applies: statsmodels documents it as a test of homogeneity on a 2x2 table (www.statsmodels.org). Dietterich found it the only test with acceptable Type I error among those compared for algorithms run once (web.engr.oregonstate.edu), and its null is that the two algorithms have the same error rate (web.engr.oregonstate.edu). Define:

- **Rewrite-induced regression** equals baseline-supported-correct requests that become incorrect or unsupported under the candidate, divided by baseline-supported-correct requests.
- **Rewrite-induced gain** is the converse, among requests that were not supported-correct at baseline.

Report both with denominators, the discordant counts, and a request-level bootstrap interval.

Step 7: decide release, fallback or clarify, using rules written beforehand

Preregister materiality rules per task type before running anything:

- **Release the rewriter** for a request class only if critical-constraint completeness meets a preregistered floor, rewrite-induced regression stays below a preregistered ceiling with its interval, and no paired-fixture family shows a systematic flip failure.
- **Fall back to original search**, or run original and rewritten searches and fuse them with the original always included, for classes that fail the floor but show a retrieval gain.
- **Clarify** where the contract marks an unresolved referent, instead of letting the rewriter choose. A rewriter that guesses silently redefines the question.
- **Publish negative outcomes and limits**, including classes where the rewriter was disabled and the denominators behind each decision.

With zero observed critical-constraint failures in n requests, the one-sided 95% upper bound on the true failure rate is approximately 3/n (the rule of three), about 1.5% at n = 200. That is arithmetic, not a finding; it shows why small fixture sets cannot certify a very low failure rate.

Data Analysis and Evidence

What the published results do and do not measure

Table 2 below summarizes reported results from the literature discussed above, with the objective each study optimized. Numbers are quoted as reported, on the datasets and metrics named, and are not transplanted to any other site.

Study	Objective or metric optimized	Reported result (as stated)	What is not reported
Ma et al. (2023), rewrite-retrieve-read	Reader answer score plus answer hit in retrieved text (aclanthology.org)	Rewriting improved most settings, except the social sciences on MMLU	Constraint preservation per request; rewrite-versus-original meaning
Wu et al. (2022), CONQRR	Retrieval reward on conversational QReCC	CONQRR (RL) outperforms the supervised T5QR rewriter by roughly 15% on MRR and Recall@10 on QReCC, for both BM25 and the dual encoder (aclanthology.org)	Rewrites are for retrievers, not end users; 34% of 50 sampled CONQRR (mix) dense-retriever errors involved misinterpretation
Wang et al. (2023), Query2doc	Retrieval metrics on MS-MARCO and TREC DL	BM25 boosted by 3% to 15%	Any audit of altered constraints
Weller et al. (2023), expansion study	nDCG across 24 retrieval models	Helps weaker, generally harms stronger models (arxiv.org)	Which constraints expansions altered
Prompt-only rewriting study (March 2026)	nDCG@10 on three BEIR sets	Minus 9.0 percent on FiQA, plus 5.1 percent on TREC-COVID (arxiv.org)	Request-level fidelity
Zheng et al., Step-Back prompting	QA accuracy; error taxonomy	PaLM-2L gains on MMLU Physics and Chemistry of 7% and 11% (iclr.cc)	Date or entity retention under abstraction in enterprise corpora; Context Loss is a catalogued error class

The last column carries the argument. Every row reports a retrieval or answer outcome, none reports the share of requests whose critical constraints survived, and the two studies closest to the question (CONQRR's error analysis and Step-Back's error taxonomy) expose meaning loss only as a minority category within a task benchmark. The directional heterogeneity, with a gain on one dataset and a loss on another, is the reason the protocol insists on site-specific measurement.

A hypothetical worked calculation

The following numbers are invented to illustrate the arithmetic (Hypothetical Example). Suppose 200 requests carry 540 applicable annotated constraints, of which 100 are critical. The candidate rewrite preserves 513 of 540 constraints, so constraint preservation is 95.0%. Yet only 164 of 200 requests preserve every critical constraint, a critical-constraint completeness of 82.0%. The average looks safe while nearly one request in five has lost a decisive qualifier.

Now suppose 120 of 200 requests are supported-correct under original search (60.0%). Under the candidate, 18 of those 120 become incorrect or unsupported, a rewrite-induced regression of 15.0%, while 30 of the other 80 become supported-correct. The candidate reaches 132 of 200 (66.0%), a net gain of six percentage points. The discordant pairs are 18 and 30; an exact two-sided McNemar test on 48 discordant pairs gives p of about 0.11, so on these invented counts the net gain is not distinguishable from zero at the 0.05 level, though a bare accuracy headline would have favored release. Whether 18 regressions are acceptable depends on the preregistered materiality rule: if 11 of the 18 involved a changed date or entity, a rule that treats those as critical would send the date-bearing class to fallback and keep the rewriter elsewhere.

Table 3 below sets out the proposed three-arm comparison scorecard. It contains no measured values, because none exist.

Measure	Original search	Human standalone reference	Candidate rewrite
Requests preserving every critical constraint	Contract applied to the unchanged search text	To be measured, with adjudication log	To be measured, per fixture family
Constraint preservation (preserved / annotated applicable)	To be measured	To be measured	To be measured
Evidence recall against original intent	To be measured	To be measured	To be measured
Supported correct answers	To be measured	To be measured	To be measured
Clarification correctness	Not applicable	To be measured	To be measured
Latency (stage and end to end)	To be measured	Not applicable	To be measured; multi-query adds retrieval calls
Measured cost per request	To be measured	Annotation hours	To be measured, including extra LLM calls

The scorecard enforces three habits. It keeps the human arm as a reference for fidelity and cost, not a production option. It reports latency and cost from measurement, because published figures vary with method and hardware; for example, the Haystack documentation describes generating a hypothetical document five times in practice (docs.haystack.deepset.ai), which multiplies calls. It also scores the original-search arm against the contract, so the baseline's own unfaithfulness, such as a follow-up question sent without its referent, is visible rather than assumed away.

Implications and Future Directions

For a team deciding whether to enable a rewriter, condenser or decomposer, the sensible default is conditional enablement. Gains reported on benchmark tasks are real for the datasets and metrics named, yet the objective those studies optimize does not include preservation of enterprise constraints, and the direction of effect varies by corpus and retriever. The LLM4CS authors acknowledge that LLM rewrite generation lacks awareness of the downstream retrieval process (ar5iv.labs.arxiv.org), so the retrieval side cannot be assumed to compensate. Enablement by request class, with fallback where critical-constraint completeness is below its preregistered floor, follows from that.

Three design consequences follow. First, carry high-risk constraints outside the free text where the architecture allows it: a date range or entity identifier applied as a metadata filter cannot be paraphrased away, although the filter values still need checking against the contract. Second, always retain the original request in the search set, since a unique-union pattern (reference.langchain.com) keeps only what the generated queries retrieve. Third, treat clarification as a first-class outcome: an evidence-bearing application with a conversational front end should ask when the referent is unresolved, as the protocol's clarification score requires.

Open questions remain. No study cited here directly measures how often LLM rewrites change a question's meaning on enterprise-style constraints; the nearest evidence is indirect, from retrieval degradation and human intent checks (ar5iv.labs.arxiv.org). Rewriters trained for one retriever or task may not transfer to another. The benefit of expansion may shrink as retrievers improve, while an LLM upstream of the retriever keeps its capacity to alter meaning. The fixtures proposed here are one candidate answer and should be revised after a first run.

Frequently Asked Questions (FAQs)

What is query rewriting in RAG?

Query rewriting in RAG is any model stage that transforms the user's request before retrieval: reformulating it, expanding it, condensing a conversation into a standalone question, or decomposing it into sub-queries (weaviate.io) (developers.llamaindex.ai). Step-back questions, which abstract a complicated query before retrieval, are one variant (milvus.io). Rewriting differs from clarification, which returns to the user.

Does query rewriting hurt RAG accuracy?

It can. Expansion generally harms stronger retrievers while helping weaker ones, and one 2026 study measured a 9.0 percent nDCG@10 fall on FiQA alongside a 5.1 percent rise on TREC-COVID. Whether it hurts a particular application is an empirical question for that corpus and retriever.

Why does a rewrite drop constraints, and which are most at risk?

Rewriters optimized for retrieval or reader scores are not rewarded for keeping qualifiers, and retrievers are weak on negation (arxiv.org), temporal relations and numeric conditions (www.sciopen.com). Negation, date relations, numeric thresholds and entity strings therefore head the intent contract.

How should dates and numeric filters be handled?

Answers to many questions depend on time, since lots of facts can evolve with respect to time (datasets-benchmarks-proceedings.neurips.cc), and dense retrievers have been reported to satisfy numerical constraints only 16.3% of the time (www.sciopen.com). Treat dates and numbers as critical constraints, test them with paired fixtures (before versus after; annual versus cumulative), and where possible apply them as structured filters outside the rewritten text. Verify the filter values against the contract, because a rewriter can still alter them.

How should rewriting be evaluated, and how is constraint preservation tested?

Annotate the contract first, compare original search, a human standalone reference and the candidate under frozen conditions, and score preservation, critical-constraint completeness, evidence recall against original intent, supported-answer correctness and clarification correctness separately. Use paired fixtures that flip one qualifier, and report paired regressions and gains with denominators and request-level resampling.

What are the common failure modes of query transformation?

Query drift (romisatriawahono.net), entity substitution (romisatriawahono.net), loss of context under abstraction, off-topic generated queries, mis-chosen conversational referents, and unions of sub-query results never checked against the original question (reference.langchain.com).

How is reformulation evaluation different from RAG evaluation?

Reformulation evaluation judges the transformed query against the contract before any retrieval. RAG benchmarks test other abilities, such as noise robustness, negative rejection, information integration and counterfactual robustness (ojs.aaai.org), and none of them is a question-fidelity test, which is why the contract is annotated separately.

Conclusion

A rewriter earns its place in an evidence-bearing application only if its retrieval gain survives an audit of what it did to the question. The primary literature shows that rewriting can raise retrieval and answer scores, and that the training objectives behind those gains target retrieval or the reader, not fidelity. It also shows that the retrievers downstream are weakest on negation, dates, numeric conditions and rare entities, so an unnoticed change is likely to travel through to a fluent, relevant and wrong answer.

The protocol proposed here asks for five things: an intent contract written before any rewrite is seen; three arms under frozen conditions; paired single-qualifier fixtures, including unresolved cases where clarification is correct and expansion cases where a non-paraphrase should pass; separate scores for the transformation, the evidence and the answer; and preregistered release, fallback and clarify rules with request-level uncertainty. It has not been run, and its thresholds are left to each team. What it prevents is the most expensive error in this area: accepting an average that hides the date, entity or polarity change that decided the answer.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://aclanthology.org/2023.emnlp-main.322.pdf
2. https://aclanthology.org/2022.emnlp-main.679
3. https://aclanthology.org/2022.emnlp-main.679.pdf
4. https://romisatriawahono.net/lecture/rm/survey/information%20retrieval/Carpineto%20-%20Automatic%20Query%20Expansion%20in%20Information%20Retrieval%20-%202012.pdf
5. https://ar5iv.labs.arxiv.org/html/2303.06573
6. https://arxiv.org/pdf/2305.07614
7. https://arxiv.org/pdf/2412.15540
8. https://www.sciopen.com/article/10.26599/BDMA.2025.9020047
9. https://arxiv.org/abs/2309.08541
10. https://arxiv.org/abs/2603.13301
11. https://web.engr.oregonstate.edu/%7Etgd/publications/dietterich-approximate-statistical-tests-nc1998.pdf

12. https://ciir-publications.cs.umass.edu/getpdf.php?id=744
13. https://www.nber.org/papers/t0344.pdf
14. https://weaviate.io/learn/knowledgecards/query-rewriting
15. https://developers.llamaindex.ai/python/framework/optimizing/advanced_retrieval/query_transformations
16. https://aclanthology.org/D19-1605
17. https://milvus.io/docs/how_to_enhance_your_rag.md
18. https://people.eng.unimelb.edu.au/ammoffat/sigir98/abstracts/mitra.html
19. https://aclanthology.org/2023.emnlp-main.585
20. https://ar5iv.labs.arxiv.org/html/2110.00768
21. https://ar5iv.labs.arxiv.org/html/2003.13624
22. https://aclanthology.org/2023.emnlp-main.322
23. https://arxiv.org/abs/2305.14283
24. https://ar5iv.labs.arxiv.org/html/2310.09716
25. https://docs.llamaindex.ai/en/stable/examples/query_engine/sub_question_query_engine
26. https://learn.microsoft.com/en-us/azure/search/agentic-retrieval-overview
27. https://reference.langchain.com/python/langchain-classic/retrievers/multi_query/MultiQueryRetriever
28. https://arxiv.org/abs/2402.03367
29. https://transacl.org/index.php/tacl/article/view/1845
30. https://iclr.cc/virtual/2024/poster/19503
31. https://arxiv.org/pdf/2310.06117
32. https://arxiv.org/abs/2305.07614
33. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/1f0e3dad99908345f7439f8ffabdffc4-Abstract-round2.html
34. https://arxiv.org/pdf/2109.08535
35. https://iclr.cc/virtual/2020/poster/1738
36. https://arxiv.org/pdf/2307.13989
37. https://ar5iv.labs.arxiv.org/html/2004.10645
38. https://ojs.aaai.org/index.php/AAAI/article/view/29728
39. https://aclanthology.org/D12-1091.pdf
40. https://www.statsmodels.org/stable/generated/statsmodels.stats.contingency_tables.mcnemar.html
41. https://docs.haystack.deepset.ai/docs/hypothetical-document-embeddings-hyde

THE PUBLISHER

About Reliable Context

Reliable Context is a system of record for organizations that fuse private and public information and must defend every value they act on.

The guarantees

One canonical record per real thing: duplicates resolve by rule, ambiguity is shown rather than hidden, and related organizations relate without ever being merged. A verbatim, cited source and an observation date sit behind every value. Unknowns are declared as not found, blocked or not applicable with a reason, never left as an empty cell. Any past state of the record is reproducible, together with what changed since.

Where it runs

It runs inside the customer's perimeter, one database per organization. Permitted sources, model routing, reader clearance and retention are set as the customer's own policies, and every value is exportable with its chain of custody for an auditor or regulator.

What it is not

It is not a vector database or a RAG framework, not a data catalog, MDM or data-quality suite, not a CRM or data-enrichment vendor, and not a chatbot or AI assistant. Where a reader needs one of those, the honest answer is to say so and point them there.

Visit

See Reliable Context.

Reliable Context is in production use in two sectors; more than six thousand of its identity decisions were made by rule and none has been reversed. Articles published here explain general practice, cite their primary and academic sources, and name the jurisdiction of any regulatory rule they describe. They are not legal, regulatory or compliance advice for a specific organization, and requirements change.

Website: https://reliablecontext.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.