# CounterRefine: Answer-Conditioned Counterevidence Retrieval for Inference-Time Knowledge Repair in Factual Question Answering

**Tianyi Huang***
Ryquo
tianyi@ryquo.com

**Ying Kai Deng**
App-In Club
kai.deng@appinclub.org

## Abstract

In factual question answering, many errors are not failures of access but failures of commitment: the system retrieves relevant evidence, yet still settles on the wrong answer. We present COUNTERREFINE, a lightweight repair layer for short-form RAG that treats the first answer as a hypothesis to test. Given a draft, COUNTERREFINE issues answer-conditioned expansion queries to retrieve candidate-specific evidence, then applies a constrained KEEP or REVISE refinement step whose proposed revisions are accepted only after deterministic validation. The design is intentionally narrow: it adds one evidence-gathering pass and one guarded refinement call rather than replacing the retriever or building a broad agentic system. On the full SimpleQA benchmark, COUNTERREFINE improves a matched one-pass RAG baseline by up to 5.8 correct-rate points; in the full Claude trace, it changes only 5.6% of outputs, with 180 beneficial outcome changes and 8 harmful ones. These findings suggest a simple but important direction for knowledgeable foundation models: beyond accessing evidence, they should also be able to use that evidence to reconsider and, when necessary, repair their own answers.

## 1 Introduction

Large language models are increasingly used as knowledge interfaces, but a central difficulty remains unresolved: a model can retrieve relevant material and still commit to the wrong fact. In short-form factual QA, these errors are unforgiving. A wrong year, nearby entity, or almost-right title is still fully incorrect. This makes factual QA a useful setting for studying a broader question that matters for knowledgeable foundation models: when an answer is wrong, can the system repair it at inference time rather than only during pre-training or parameter editing?

Retrieval-augmented generation (RAG) improves factuality by grounding generation in external evidence rather than relying only on parametric memory (Lewis et al., 2021; Karpukhin et al., 2020). Yet retrieval alone does not eliminate candidate-selection errors. A first-pass retriever is optimized for *topic relevance*, not necessarily for *candidate discrimination*. Once a draft answer exists, however, the retrieval problem changes. A useful next query may include the candidate answer itself: if the draft year is wrong, adding that year to the query can surface evidence that falsifies it; if the draft entity is ambiguous, answer-conditioned re-querying can reveal a more discriminative title, list entry, or sentence.

We build on this observation with COUNTERREFINE, a retrieval-based short-answer system that first drafts an answer from web snippets and then performs one answer-conditioned evidence-gathering pass. This second stage is best viewed as a focused form of query expansion, not as a new general-purpose retrieval algorithm. Its role is to expose evidence that bears directly on a concrete hypothesis. A constrained refinement model then decides whether to KEEP the draft or REVISE it, and a deterministic validator blocks unsupported or ill-formed rewrites. The result is a simple repair layer that can sit on top of an existing retrieval pipeline.

Our empirical evidence focuses on matched evaluation. On the full 4,326-question SimpleQA benchmark with the official grader (Wei et al., 2024; OpenAI, 2025), COUNTERREFINE improves correct rate by +4.0 points with Claude Sonnet 4.6 and +5.8 points with GPT-5 over the corresponding one-pass retrieval baselines. We also performed preliminary transfer checks on 100 and 300 sample slices of HotpotQA distractor development (Yang et al., 2018), where the exact match was improved for both backbones. These results support a bounded claim: answer-conditioned evidence gathering can be useful when it is paired with a conservative gate,

*Corresponding author.

but it is not a replacement for broader verification systems or stronger retrieval.

Our contributions are: (1) we instantiate answer-conditioned query expansion as a candidate-testing step inside a guarded short-answer repair layer; (2) we show that the complete layer improves over a matched retrieval baseline under the official SimpleQA evaluation pipeline; and (3) we provide an intervention profile and representative examples that clarify when the method helps and when it can still fail.

## 2 Related Work

**Retrieval-grounded factual QA.** Retrieval-augmented generation improves knowledge-intensive NLP by grounding generation in external documents (Lewis et al., 2021). Dense Passage Retrieval further improved open-domain QA with learned neural retrieval (Karpukhin et al., 2020). More recent work has explored adaptive retrieval and critique, for example by teaching models when to retrieve and how to reflect on retrieved evidence (Asai et al., 2024). Other work has targeted robustness to noisy retrieval (Yu et al., 2024) or refreshed model knowledge through search-engine augmentation (Vu et al., 2024). Our method does not redesign the retriever or the full RAG stack; it adds a small second-pass evidence-gathering step that is conditioned on the first answer.

**Query expansion and answer-conditioned retrieval.** The retrieval step in COUNTERREFINE is closely related to query expansion. Classical query expansion and pseudo-relevance-feedback methods have long expanded queries using terms inferred from local document evidence, global corpus statistics, or estimated relevance models (Xu and Croft, 1996; Lavrenko and Croft, 2001; Carpineto and Romano, 2012). Recent LLM-based variants generate auxiliary text for retrieval, including HyDE, which retrieves through hypothetical document embeddings (Gao et al., 2022), and Query2Doc, which expands queries with LLM-generated pseudo-documents (Wang et al., 2023). COUNTERREFINE shares the broad idea of using auxiliary generated text to improve retrieval. Its focus is narrower: the auxiliary text is the system's own draft answer, and the expanded retrieval pass is embedded in a conservative `KEEP/REVISE` repair layer for short factual answers.

**Verification and self-correction.** A broad line of work improves factuality by checking or revising an initial answer. Chain-of-Verification generates verification questions before producing a final answer (Dhuliawala et al., 2024). CRITIC uses tool-interactive critique to validate and amend outputs (Gou et al., 2024). RARR retrieves evidence and revises unsupported claims while trying to preserve the original text (Gao et al., 2023). SelfCheckGPT detects likely hallucinations through disagreement across samples (Manakul et al., 2023). COUNTERREFINE is closest in spirit to this family. We do not claim that it dominates these broader systems; rather, we study a deliberately constrained setting: one additional evidence-gathering phase, one refinement call, and a short-answer decision with deterministic validation.

**Knowledge repair and model editing.** Wrong knowledge can also be addressed by editing model parameters directly. ROME and MEMIT show that factual associations in language models can be modified through targeted weight updates (Meng et al., 2022, 2023). Those methods operate on the model's stored knowledge itself. COUNTERREFINE is complementary: it performs answer-level repair at inference time instead of changing model parameters, using external evidence to challenge and revise a candidate answer.

**Benchmarks and conflicting evidence.** Factuality has been studied through claim verification datasets such as FEVER (Thorne et al., 2018), factual QA benchmarks such as TruthfulQA (Lin et al., 2022), long-form factuality metrics such as FACTSCORE (Min et al., 2023), and broader surveys of LLM factuality (Wang et al., 2024). SimpleQA is particularly relevant here because it isolates short-form factual precision in a benchmark of 4,326 questions with a single indisputable answer (Wei et al., 2024). Recent work has also emphasized that retrieval errors are not limited to irrelevant documents; systems often face ambiguous, noisy, or conflicting evidence (Wang et al., 2025). Our setting is simpler than that literature's multi-document conflict benchmarks, but the motivating issue is related: even in short-answer QA, retrieved snippets can weakly support multiple nearby candidates. Answer-conditioned re-querying is one lightweight way to make those conflicts visible before finalizing an answer.

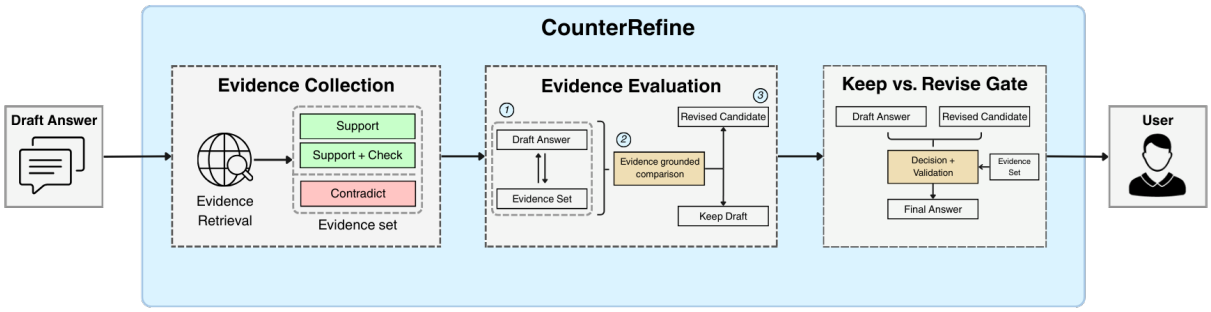


Figure 1: Schematic overview of COUNTERREFINE. Starting from a draft answer produced by BASELINE-RAG, COUNTERREFINE collects additional evidence intended to support, check, or contradict that candidate, evaluates the draft answer against the resulting evidence set, and then passes the result to a constrained keep-vs.-revise gate. The final answer is either the original draft or a revised candidate accepted under evidence-based validation using the same evidence set. For readability, the figure abstracts away some implementation details from the text, including question-type-conditioned query selection and per-query snippet aggregation.

## 3 Method

### 3.1 Overview

We refer to the first-pass retrieval system as BASELINE-RAG. Given a question $q$, COUNTERREFINE first runs BASELINE-RAG to produce an initial answer $a_0$, then runs one answer-conditioned refinement stage:

$$R_0 = \mathrm{Retrieve}(q, k_b), \tag{1}$$
$$a_0 = f_{\text{draft}}(q, R_0), \tag{2}$$
$$R_1 = \mathrm{RetrieveRefine}(q, a_0, R_0), \tag{3}$$
$$a^\star = f_{\text{refine}}(q, a_0, R_1). \tag{4}$$

The pipeline is designed as a focused reliability layer on top of an existing retrieval system.

### 3.2 Stage 1: retrieval-based drafting

The baseline stage retrieves up to $k_b$ evidence snippets for the original question:

$$R_0 = \mathrm{Retrieve}(q, k_b). \tag{5}$$

For the SimpleQA web-retrieval experiments, $k_b = 5$. For the HotpotQA component ablation, the local distractor-context runner keeps up to 10 first-pass paragraphs. The retriever returns tuples of title, URL, and evidence text when URLs are available, and duplicates are removed before prompt construction.

The drafting model answers using only the retrieved evidence:

$$a_0 = f_{\text{draft}}(q, R_0). \tag{6}$$

The draft prompt enforces a short answer and disallows hedging or explanation. If retrieval fails to yield a valid answer, the code falls back to a closed-book short-answer prompt; if that also fails, a minimal type-dependent default is emitted.

### 3.3 Stage 2: answer-conditioned query expansion

The refinement queries depend on the draft answer. Let $t(q)$ denote a coarse question type inferred from the question string. The refinement query set is

$$Q(q, a_0) = \{q,\ q\|a_0\} \cup \begin{cases} \{a_0\}, & t(q) \in \mathcal{T}, \\ \varnothing, & \text{otherwise} \end{cases} \tag{7}$$

where $\|$ denotes concatenation of the question and the surface form of the draft answer with a separator, and

$$\mathcal{T} = \{\texttt{who}, \texttt{where}, \texttt{when}, \texttt{year}, \texttt{number}\}.$$

Thus, every example receives an answer-conditioned query containing both the original question and the draft answer, and slot-like questions also receive a bare-answer query. This is intended to retrieve evidence that directly bears on the provisional answer, including evidence that may support, sharpen, or contradict it.

The question-type heuristic is intentionally simple. It labels yes/no questions when the question begins with an auxiliary verb; year questions when the question contains cues such as *what year* or *in what year*; temporal questions when it begins with *when* or asks for a month, date, or timeframe; numeric questions when it contains cues such as *how many*, *how much*, *population*, or *number*; location questions when it begins with *where* or asks for a city, county, town, village, municipality, or

| Step | Example trace |
|---|---|
| Question | Who was awarded the Oceanography Society's Jerlov Award in 2018? |
| Draft | Collin Roesler |
| Second-pass queries | Original question; original question + "Collin Roesler"; "Collin Roesler" |
| Retrieved signal | Award-list evidence names Annick Bricaud as the 2018 recipient. |
| Decision | REVISE to Annick Bricaud |

Table 1: Illustrative Stage-2/Stage-3 trace. The second pass does not solve the question from scratch; it tests a concrete draft answer against newly surfaced evidence.

neighborhood; and person questions when it begins with *who*, *whom*, or *whose*. All other questions are assigned to a residual `other` type.

For each $q' \in Q(q, a_0)$, the system retrieves up to $k_r$ evidence snippets, with $k_r = 5$ in the current implementation, and merges them with the baseline support set:

$$R_1 = \mathrm{Dedupe}\Big(R_0 \cup \bigcup_{q' \in Q(q,a_0)} \mathrm{Retrieve}(q', k_r)\Big). \tag{8}$$

Table 1 gives a compact example. The first-pass answer is plausible but wrong. Conditioning retrieval on the draft answer surfaces a more discriminative award-list signal, after which the refinement gate can revise the answer.

### 3.4 Stage 3: constrained refinement

The refiner receives the question, the baseline answer, and the merged evidence set, and must output exactly three fields:

```
DECISION: KEEP or REVISE
ANSWER: <short answer only>
EVIDENCE: <one short evidence snippet or
NONE>
```

The prompt instructs the model to `KEEP` when the baseline is already the best supported answer and to `REVISE` only when the evidence strongly supports a different or more exact short answer. The prompt does not ask for a chain of reasoning; the evidence field is used only as a short grounding signal for validation.

Let $(d, \hat{a}, e)$ denote the parsed refinement output. If $d = \texttt{KEEP}$, the system returns $a_0$. If $d = \texttt{REVISE}$, the proposed answer must pass deterministic validation before it is accepted. Malformed outputs, missing required fields, empty answers, and non-answer strings are treated conservatively: the system falls back to the draft rather than emitting an unvalidated rewrite.

### 3.5 Deterministic validation and canonicalization

The validator blocks revisions that are unsupported, ill-typed, or stylistically incompatible with short-answer evaluation. A proposed rewrite is rejected if any of the following hold:

1. it is empty, non-responsive, or identical to the draft after normalization;
2. for yes/no questions, it is not exactly `yes` or `no`;
3. for entity-style questions, it is clause-like, overly long, or begins with a descriptor phrase;
4. for temporal or numeric questions, it lacks an explicit temporal or numeric marker;
5. the refiner provides no evidence snippet;
6. lexical grounding between the revised answer and the cited evidence is too weak.

Rule 6 is implemented as a necessary grounding check after lowercasing and punctuation removal. For temporal and numeric questions, at least one year, digit-bearing token, or month token from the proposed answer must appear in the cited evidence. For entity-style answers, at least one normalized non-stopword answer token must appear in the cited evidence; for multi-token answers, the check favors contiguous or near-contiguous overlap when available. This rule is intentionally conservative: it cannot prove that a revision is correct, but it blocks revisions that are not even lexically anchored in the evidence supplied by the refiner.

Accepted revisions are then canonicalized with question-type-specific rules, such as extracting a 4-digit year, compacting number spans, stripping leading prepositions for locations, or removing parenthetical descriptors from names. Only proposed revisions are validated; `KEEP` decisions preserve the baseline answer unchanged. The final output is

$$a^\star = \begin{cases} \mathrm{Canon}(q, \hat{a}), & \text{if the revision is accepted,} \\ a_0, & \text{otherwise.} \end{cases} \tag{9}$$

## 4 Experimental Setup

### 4.1 Benchmarks and metrics

We evaluate on two benchmarks. Our primary benchmark is SimpleQA, which contains 4,326 short, fact-seeking questions with a single indisputable answer (Wei et al., 2024). We use

the official SimpleQA grader from the public `simple-evals` implementation (OpenAI, 2025). Following the official setup, we report the correct rate and F1 score.

As a preliminary transfer setting, we evaluate on 100-example and 300-example slices from the HotpotQA distractor development set (Yang et al., 2018). HotpotQA is useful here because it is more multi-hop and span-oriented than SimpleQA. We report the standard exact match (EM) and token-overlap F1 metrics. We also run a 300-example HotpotQA component ablation with the same backbone within each run. This ablation is intended for within-configuration attribution rather than as a new benchmark-level HotpotQA claim.

### 4.2 Systems and decoding

Our primary full-benchmark run uses Claude Sonnet 4.6 for both drafting and refinement (Anthropic, 2026). Retrieval is implemented through the OpenAI web search API in a simple RAG pipeline that returns web evidence for subsequent answer generation (OpenAI, 2026; Lewis et al., 2021). The baseline system, BASELINE-RAG, uses the same retrieval component and the same drafting backbone but does not run the answer-conditioned refinement stage. This keeps the comparison focused on the contribution of COUNTERREFINE rather than on backbone or first-pass retrieval differences.

We also ran matched experiments with GPT-5 on SimpleQA and HotpotQA (OpenAI, 2025). All reported numbers are single-run benchmark results under the same retrieval and prompting configuration within each backbone. Where the provider API exposes a temperature or comparable stochasticity control, we use a low-randomness setting. We do not report multi-seed variance; this is one reason we emphasize the full SimpleQA matched-baseline comparison over small-slice differences.

### 4.3 Computational profile

Table 2 reports per-question operation counts for the SimpleQA web-retrieval setting before deduplication. We report counts rather than dollar cost because provider pricing, caching, and latency vary by model, date, and deployment environment. The main overhead is bounded and explicit: COUNTERREFINE adds one refinement model call and two or three second-pass retrieval queries, depending on question type.

| System | Retrieval queries | Raw snippets | Model calls |
|---|---|---|---|
| BASELINE-RAG | 1 | $\leq 5$ | 1 draft |
| COUNTERREFINE | 3–4 | $\leq$ 15–20 | 1 draft + 1 refine |

Table 2: Per-question computational profile for the SimpleQA web-retrieval setting before deduplication. The 3–4 retrieval-query range for COUNTERREFINE includes the shared first-pass query and depends on whether the question type triggers the bare-answer query.

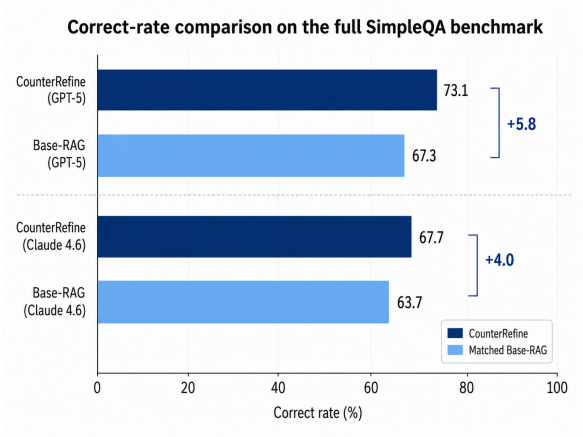


Figure 2: Correct-rate comparison on the full SimpleQA benchmark. COUNTERREFINE is compared only with the matched one-pass retrieval baseline using the same backbone and retrieval setup.

## 5 Results

### 5.1 Performance across benchmarks and backbones

Table 3 reports the main quantitative results. Adding COUNTERREFINE on top of BASELINE-RAG improves the primary exactness-oriented metric in each reported setting. The strongest evidence is the full SimpleQA evaluation, where both systems use the same retriever and backbone. The smaller SimpleQA and HotpotQA slices are useful checks, but they should not be read as establishing broad benchmark-level superiority.

Two aspects of the result are worth noting. First, the improvement appears on top of an already retrieval-grounded baseline rather than a no-retrieval model. The comparison is therefore between a matched one-pass retrieval pipeline and the same pipeline augmented with answer-conditioned refinement. Second, the gains are strongest on metrics that reward final answer exactness. This is most visible in the SimpleQA correct rate and Hot-

| Benchmark | Metric | Claude 4.6 | | | GPT-5 | | |
|---|---|---|---|---|---|---|---|
| | | Base-RAG | COUNTERREFINE | Δ | Base-RAG | COUNTERREFINE | Δ |
| SimpleQA (100) | Correct↑ | 65.0 | **75.0** | +10.0 | 57.0 | **71.0** | +14.0 |
| | F1↑ | 65.0 | **75.0** | +10.0 | 59.7 | **73.2** | +13.5 |
| SimpleQA (4,326) | Correct↑ | 63.7 | **67.7** | +4.0 | 67.3 | **73.1** | +5.8 |
| | F1↑ | 64.1 | **68.1** | +4.0 | 58.6 | **72.1** | +13.5 |
| HotpotQA (100) | EM↑ | 74.0 | **79.0** | +5.0 | 71.0 | **76.0** | +5.0 |
| | F1↑ | 82.2 | **84.9** | +2.7 | 83.7 | **84.5** | +0.8 |
| HotpotQA (300) | EM↑ | 70.0 | **76.0** | +6.0 | 68.0 | **71.7** | +3.7 |
| | F1↑ | 83.9 | **86.1** | +2.2 | 83.6 | **84.9** | +1.3 |

Table 3: Main results across backbones and benchmarks. BASELINE-RAG is the matched one-pass retrieval baseline using the same retriever and backbone as COUNTERREFINE. SimpleQA uses the official grader. HotpotQA uses the distractor development setting on 100-example and 300-example slices. The HotpotQA (300) COUNTERREFINE values are the full-system row of Table 4. All values are percentages.

potQA EM, which matches the intended role of COUNTERREFINE as a repair layer for final factual precision rather than a general-purpose reasoning scaffold.

For GPT-5 on full SimpleQA, the F1 gain is larger than the correct-rate gain. We interpret this cautiously. F1 is sensitive to answer surface form and tokenization, while COUNTERREFINE canonicalizes accepted revisions into shorter slot-like answers. We therefore treat official SimpleQA correct rate as the primary metric and F1 as a secondary surface-overlap diagnostic.

The cross-benchmark pattern is also informative but preliminary. On SimpleQA, where a near-miss entity, date, or number is scored as fully wrong, answer-conditioned refinement yields clear gains under the official evaluation pipeline. On HotpotQA, the same mechanism improves exact match on the reported slices, including the 300-example runs aligned with the component ablation. Because these HotpotQA runs cover slices rather than the full development set, we use them as evidence of transfer potential rather than as a complete characterization of multi-hop performance.

### 5.2 Component ablations on HotpotQA

Table 4 reports a 300-example HotpotQA component ablation. The rows compare the full system with four nearby variants: a second pass that reuses only the original question, answer-conditioned refinement without deterministic validation, answer-conditioned refinement without canonicalization, and a simple reconsideration baseline with the same additional model-call budget. The Full COUNTERREFINE row is the same HotpotQA-300 run reported in Table 3, using the same matched baseline. These rows are meant to compare variants within a fixed local distractor-context setup; they are not intended to replace the full SimpleQA evaluation as the main evidence for the method.

The ablation results are useful but limited. Relative to the matched HotpotQA-300 baseline in Table 3, the full system improves Claude 4.6 by +6.0 EM and +2.2 F1, and GPT-5 by +3.7 EM and +1.3 F1. The nearby variants fall in a narrow range, so this slice does not isolate a large independent contribution from any single component. We use the ablation as a within-configuration check on the second-pass repair family; the strongest evidence for the guarded design remains the full SimpleQA matched-baseline gain and selective-intervention profile.

### 5.3 Intervention profile

The full Claude SimpleQA trace helps explain the matched-baseline gain. COUNTERREFINE changes only a small fraction of predictions, and those changes are mostly beneficial: the system revises 5.6% of examples, helps 180, hurts 8, and produces zero refiner-format failures. Beneficial outcome-changing revisions therefore outnumber harmful ones by 22.5 to 1. Here, a refiner-format failure means an API/runtime failure or malformed output that prevents parsing the required fields. Validator-rejected `REVISE` proposals are handled conservatively by returning the draft and are separate from this failure count. The not-attempted rate remains fixed at 1.2%, so the improvement is not driven by abstaining more often.

Figure 3 makes the structure of the gain explicit.

| Variant | Claude 4.6 | | GPT-5 | |
|---|---|---|---|---|
| | EM↑ | F1↑ | EM↑ | F1↑ |
| Full COUNTERREFINE | 76.0 | 86.1 | 71.7 | 84.9 |
| Second pass with original question only | 76.0 | 86.1 | 71.7 | 84.9 |
| Answer-conditioned, no validator | 74.7 | 86.0 | 72.0 | 85.0 |
| Answer-conditioned, no canonicalization | 75.0 | 86.5 | 71.7 | 84.9 |
| Simple reconsideration | 74.7 | 86.0 | 72.0 | 85.0 |

Table 4: HotpotQA-300 component ablation. The table compares the full COUNTERREFINE pipeline with variants that use only the original question in the second pass, remove deterministic validation, remove answer canonicalization, or replace the guarded refinement step with simple reconsideration. Values are percentages; the Full COUNTERREFINE row corresponds to the HotpotQA-300 system row in Table 3, and EM is rounded from exact-match counts over 300 examples.

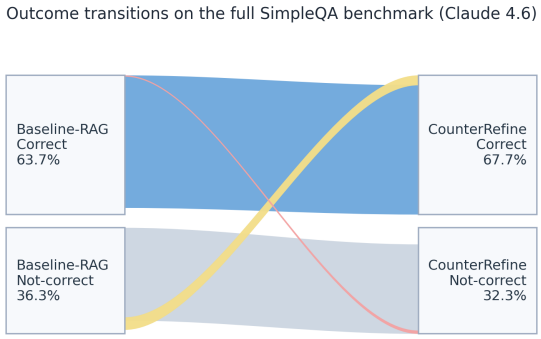


Figure 3: Outcome transitions on the full SimpleQA benchmark for Claude 4.6. Of all examples, 63.5% stay correct, 32.1% stay not-correct, 4.2% are corrected (incorrect → correct), and 0.2% are harmed (correct → incorrect).

Most examples are unchanged, but among changed outcomes, incorrect → correct transitions dominate correct → incorrect ones by a wide margin. This is the intended behavior for a post-retrieval repair layer: it should intervene selectively, and when it does, it should usually move the answer in the right direction.

### 5.4 Representative failures and successes

Table 5 illustrates the main failure and success patterns. The hurt cases are not random; they are mostly cases of *relation confusion* or *event mismatch*. In these examples, the refinement stage retrieves evidence about a nearby but different relation, milestone, or date, and that adjacent fact is plausible enough to override an initially correct answer. This suggests that the main remaining weakness is not lack of evidence, but insufficient discrimination between closely related factual relations.

The helped cases show the complementary pattern. BASELINE-RAG is often nearly correct but too coarse, under-specified, or anchored on the wrong nearby candidate. Once the draft answer is injected back into retrieval, the second-stage search can surface a decisive snippet that the first pass failed to isolate. The benefit is therefore not an unconstrained increase in reasoning effort; it is a change in what evidence becomes visible after a concrete hypothesis has been formed.

## 6 Discussion

The results support a bounded interpretation: one-pass retrieval sometimes fails not because relevant evidence is globally unavailable, but because the system commits to a plausible candidate before retrieving evidence that would better test that candidate. COUNTERREFINE addresses this failure mode by changing the role of retrieval after drafting. The second retrieval pass is no longer only about finding topic-relevant material; it is also about checking a concrete factual hypothesis.

This framing also clarifies the scope of the claim. The answer-conditioned retrieval step is a form of query expansion, and Table 3 evaluates the complete repair layer rather than query expansion alone. The HotpotQA ablations in Table 4 support a cautious reading: nearby second-pass variants perform similarly on that slice, so we do not claim that each component contributes a large independent gain in every setting. The main empirical claim is more modest: in short-form factual QA, a matched RAG system can benefit from a lightweight second-pass repair step when candidate-specific retrieval is paired with a conservative keep-or-revise gate.

| Question (abridged) | BASELINE-RAG | Final | Gold | Diagnosis |
|---|---|---|---|---|
| **Representative hurt cases** | | | | |
| Who appointed the Chief Justice of India, Mirza Hameedullah Beg, in 1977? | Fakhruddin Ali Ahmed | Indira Gandhi | Fakhruddin Ali Ahmed | Nearby evidence named the government behind the elevation rather than the appointing president requested by the question. |
| What day, month, and year was the municipality of Arboletes, Antioquia, Colombia, founded? | 1920 | August 1958 | July 20, 1920 | The refinement evidence referred to municipal establishment rather than the original founding date. |
| On what day, month, and year did Manny Pacquiao marry Jinkee Jamora? | May 10, 1999 | May 10, 2000 | May 10, 1999 | A noisy timeline-style snippet overrode an already correct baseline year. |
| **Representative helped cases** | | | | |
| Who was awarded the Oceanography Society's Jerlov Award in 2018? | Collin Roesler | Annick Bricaud | Annick Bricaud | Answer-conditioned retrieval surfaced an award-list entry explicitly naming the 2018 winner. |
| What were the day, month, and year of death of Mehr Chand Mahajan? | 5 December 1984 | 11 December 1967 | 11 December 1967 | The refinement stage found a biographical snippet containing the exact parenthetical death date. |
| In which year did Fazal Ilahi Chaudhry join the Muslim League? | 1940s | 1942 | 1942 | Re-querying with the draft answer surfaced the exact year instead of a decade-level answer. |

Table 5: Representative failures and successes from the full Claude trace log.

## 7 Conclusion

We presented COUNTERREFINE, a retrieval-based short-answer QA system that answers first and then retrieves evidence targeted at its own provisional answer. Across the reported evaluations, the method improves a matched one-pass RAG baseline, with the strongest evidence coming from full SimpleQA evaluation under the official grader and the HotpotQA results serving as preliminary transfer checks. The results suggest a simple repair pattern: treat the first answer as a hypothesis, retrieve evidence that tests that hypothesis, and accept revisions only when they pass conservative validation.

## Limitations

The empirical evidence is strongest for short-form factual settings, particularly SimpleQA, where exact entity, date, and value corrections are central to evaluation. The HotpotQA results are positive but cover limited slices rather than the full benchmark; they should be interpreted as preliminary evidence of transfer potential, not as a complete characterization of multi-hop performance.

Our primary comparison is a matched one-pass retrieval baseline. This isolates the value of adding COUNTERREFINE to the same retriever and backbone, but it does not establish superiority over broader verification and editing systems such as RARR, CRITIC, Chain-of-Verification, or agentic tool-use pipelines. Our component ablations are limited to a 300-example HotpotQA slice and show small differences among nearby variants. They are useful as a diagnostic check, but they do not fully attribute the SimpleQA gains among answer-conditioned retrieval, deterministic validation, and canonicalization.

Finally, COUNTERREFINE is designed to repair a draft answer after retrieval, not to replace retrieval itself or to solve long-form generation problems. Its effectiveness depends on whether retrieval surfaces evidence that is informative enough for the validator and refiner. We have not yet evaluated smaller refinement models or latency-optimized deployment variants, which are natural directions for reducing cost.

## Broader Impact and Ethical Considerations

COUNTERREFINE points to a practical direction for improving retrieval-grounded QA systems: before an answer reaches a user, the system can use additional evidence to test whether a plausible draft should be kept or revised. Because the method is modular and compatible with standard retrieval pipelines, it may be useful as a reliability layer for search assistants, educational tools, and scientific interfaces where small factual errors can matter.

At the same time, inference-time repair can improve evidential support, but it does not guarantee truth. If retrieved evidence reflects social, historical, or institutional bias, a repair layer may reinforce those patterns rather than correct them. In high-stakes settings, repaired answers should therefore be treated as better-supported outputs rather than as definitive verification. Systems like COUNTERREFINE should support, not replace, human judgment, especially in domains such as medicine, law, public policy, and education.

## References


Anthropic. 2026. Introducing claude sonnet 4.6.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *Preprint*, arXiv:2212.10496.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127. Association for Computing Machinery.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfcheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5 is here.

OpenAI. 2025. simple-evals. GitHub repository. Archived benchmark results table and reference implementation; accessed March 12, 2026.

OpenAI. 2026. Web search. https://developers.openai.com/api/docs/guides/tools-web-search. OpenAI API documentation.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. Fresh-LLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720. Association for Computational Linguistics.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-augmented generation with conflicting evidence. *Preprint*, arXiv:2504.13079.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *Preprint*, arXiv:2303.07678.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Nenkov Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024. Factuality of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19519–19529. Association for Computational Linguistics.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, page 4–11. Association for Computing Machinery.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685. Association for Computational Linguistics.