

Season 2: Retrieval and RAG

Rewriting the question.

One cheap model call before search: what it fixes, what it cannot, and what it quietly breaks. Three ways to rewrite, tested against the 520 test questions.

Search only sees the last message

1 Load	2 Chunk	3 Embed	4 Rewrite	5 Search	6 Filter	7 Rerank	8 Prompt	9 Answer	10 Cite	11 Score
-----------	------------	------------	--------------	-------------	-------------	-------------	-------------	-------------	------------	-------------

● ● ● Halvard Home support chat

a real test conversation

How long should I wait to turn on my Orrin heat pump after moving it?

You

H

After moving the Orrin heat pump, wait 9 hours before switching it on.

What about the Mira robot vacuum?

You

WHAT SEARCH RECEIVES

"What about the Mira robot vacuum?"

The first two messages are not sent

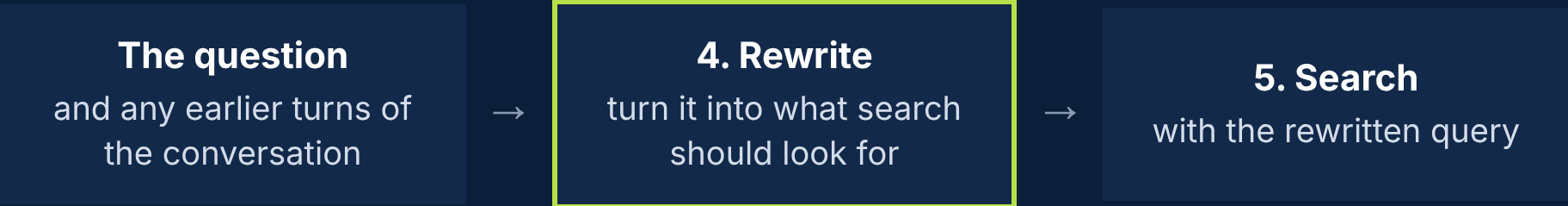
"What about the Mira robot vacuum?" names a product, but not **what** is being asked. Search ranked the answer **10th**. Across 40 follow-ups like this, it found the answer **7 times**.

The question

If a model rewrites the question before search, does search find more? And what does it cost?

Where stage 4 sits

Stage 4 runs live, on every question, just before search.



Until now	Stage 4 has passed the question straight through. Search has always seen exactly what the customer typed last.
What changes	Only the search query . The model that writes the answer still sees the customer's own words and the whole conversation.
The price	One extra call to a small, cheap model, before search can start.
Why it can help	Search matches the query against chunks (E5). A query that says what the customer actually means gives search something to match.

Three kinds of question search fails

All three are in Halvard's test set. Search found the answer for 7 of the 80, as typed.

Follow-up

40 questions · 7 found

Leans on earlier turns for its meaning.

"What about the Mira robot vacuum?"

Two-part

20 questions · 0 found

Asks two things at once, answered in two different passages. Both must be in the top 5.

"How soon can an engineer visit in Wales and West, and how much is a Brio kettle service?"

Bridge

20 questions · 0 found

Needs one fact to find another. The passage with the answer never mentions what the customer said.

"How long do repair parts take to arrive in postcode area LL?"

A rewrite can supply missing context, or break a question into parts. Can it bridge? Hold that thought.

Three ways to rewrite

Each is one call to gpt-6-luna with a short instruction. Abridged:

Rewrite

every question

"Turn a customer's question into a search query. Keep every product name, code and number exactly. Add the words the documents are likely to use."

Split

every question

"If the question asks for more than one thing, write each part as a stand-alone question. Otherwise repeat it." Each part is searched, and the lists are fused as in E5.

Resolve

follow-ups only

Called only when there are earlier turns. "Rewrite the last turn as a stand-alone question. Bring in what it refers to from the conversation."

All three share one rule

Keep every product name, code and number exactly as written. Watch what happens to codes anyway.

Stage 4: Rewrite the question

Slide 5 described three ways to rewrite. This slide shows how stage 4 runs each one; the next applies all three to a real question.

WHAT CHANGES IN THIS STAGE

Before search, a small model reads the conversation and the latest question, and replies with the query search should use. The same code runs all three ways; only the mode, and so the instruction, changes. Each call is about 140 tokens: under a hundredth of a cent.

```
E5 rewrite: {mode: passthrough}
```

```
Way 1 rewrite: {mode: rewrite, model: gpt-6-luna}
```

```
Way 2 rewrite: {mode: split, model: gpt-6-luna}
```

```
Way 3 rewrite: {mode: resolve_followup, model: gpt-6-luna}
```

src/lab/rag/rewrite.py

the whole function

```
def rewrite_prompt(item):
    """What the rewriting model reads: any earlier turns,
    then the last one. The mode's instruction is the system
    prompt."""
    parts = []
    if item.history:
        parts.append(conversation(item.history))
    parts.append(f"Question: {item.question}")
    return "\n\n".join(parts)
```

The follow-up, three ways

The three ways from slide 5, each given the same conversation.

AFTER ASKING ABOUT THE ORRIN HEAT PUMP, THE CUSTOMER TYPES
What about the Mira robot vacuum? Answer: wait 37 hours

WHAT SEARCH RECEIVED, AND WHERE THE ANSWER RANKED

As typed the reference	What about the Mira robot vacuum?	10th
1. Rewrite	Mira robot vacuum wait after moving before turning on relocation settling time	2nd
2. Split	How long should I wait to turn on my Mira robot vacuum after moving it?	2nd
3. Resolve	How long should I wait to turn on my Mira robot vacuum after moving it?	2nd

What the model did
It carried "how long should I wait after moving it" across from the earlier turn. Search now knows what is being asked about the vacuum, and finds it. The answer goes from wrong to right.

Note the two styles: rewrite writes a list of search words; split and resolve write the question a person would ask. With one question to ask, split returns it whole.

Our question, rewritten

OUR QUESTION

How much longer does Extended Care keep my warranty going? Answer: 58 months

As typed	How much longer does Extended Care keep my warranty going?	2nd
Rewrite	Extended Care warranty length extended warranty coverage duration additional years warranty period	2nd
Split, Resolve	unchanged: one part, and no earlier turns	2nd

Nothing to fix, nothing broken

Our question already says what it means, so every mode finds the answer 2nd. Rewrite still made a call and still changed the query. On questions like ours, that is cost for no gain. Remember that for slide 13.

The test: four versions of stage 4

Every result from here on compares these four. Nothing else in the system changes.

As typed

the reference

No rewriting. Today's system.

Rewrite

Every question becomes a search query.

Split

Every question split into parts, if it has any.

Resolve

Follow-ups only become stand-alone questions.

1 Round 1: search only, all four

All 400 answerable questions. The rewrite calls are so cheap that every mode runs.

2 Round 2: the full system, one mode

Chosen on the dev questions: rewrite found 10 of 20 hard ones, split 9, resolve 8. Rewrite ran end to end on all 520.

The rule to join, set in advance

Search clearly above the reference's (the likely ranges must not overlap), and right answers not lower.

Follow-ups fixed. Bridges untouched.

Search: right passage in the top 5, by kind of question.

	FOLLOW-UP OF 40	TWO-PART OF 20	BRIDGE OF 20	CODES OF 40	ALL 400
As typed the reference	7	0	0	15	281
Rewrite	35	0	0	7	296
Split	34	1	0	15	309
Resolve	34	0	0	15	308

Follow-ups	From 7 to about 35, whichever mode. Supplying the missing context works.
Two-part, bridge	Still almost nothing. The next two slides show why.
Codes	Rewrite halved them, 15 to 7. Slide 13.

Split finds both halves, but not together

TWO-PART QUESTION

In Wales and West, how soon can a warranty engineer visit after my claim is accepted, and how much does a routine service for the Brio kettle cost?

SPLIT'S TWO QUERIES

Part 1

In Wales and West, how soon can a warranty engineer visit after my claim is accepted?

Part 2

How much does a routine service for the Brio kettle cost?

Each query looks for one passage, a far easier search. But the two lists are then fused into one top 5, and here the second passage landed **6th**, one place short.

The lesson

A two-part question needs **both** passages in front of the model. Splitting fixes the search, but five places shared between two lists is not enough room. More passages (a bigger k, E9) or one slot per part would be the next thing to try. Split found both for 1 of 20.

A bridge cannot be rewritten

BRIDGE QUESTION

How long do parts for warranty repairs take to arrive where I live, in postcode area LL? Answer: 57 working days

THE ANSWER NEEDS TWO PASSAGES, FOUND IN ORDER

Step 1: which region?

"The Wales and West service team covers postcode areas CF, SA, LL, BS and GL."

→

Step 2: that region's time

"Parts for warranty repairs in Wales and West arrive within 57 working days."

The passage with the answer says "Wales and West", never "LL". To search for it, you must already know the region. The rewriting model does not.

Split's query

How long do parts for warranty repairs take to arrive in postcode area LL?

30th

0 of 20, every mode

A bridge needs search, read, then search again with what was learned. That loop is agentic retrieval, in Season 5.

Rewriting everything broke the codes

FROM E5
What does error E-2194 mean? Vector search ranked it 10th

As typed	What does error E-2194 mean?	10th
Rewrite	Halvard Home error code E-2194 meaning troubleshooting	17th

The model kept the code, exactly as told. But it added generic words that every error article shares: "Halvard Home", "error code", "troubleshooting". To an embedding, the query now looks even more like **every** error article.

Codes	15 found as typed, 7 after rewriting.
Everything else	The 320 questions that were not follow-ups, two-part or bridge: 274 as typed, 261 rewritten. Not a clear loss, but not free.

The lesson
A rewrite that helps the questions that need it can hurt the ones that did not. Rewrite only what needs it.

Better search, better answers

Round 2: rewrite against as typed, the whole system, all 520 test questions.

	SEARCH	RIGHT	NOT FOUND	LEAKS
As typed the reference	281 of 400	280 of 400	42 of 120	76 of 480
Rewrite	296 +15	296 +16	43 +1	98 +22

Search to answers

15 more passages found, 16 more right answers. As in E4, when search finds more, the model answers more.

Leaks

Up again, to 98. Nothing stops staff-only text reaching a customer's prompt yet; E8's filter does.

Split and resolve found even more in round 1, 309 and 308. They were not run in full: the dev questions chose rewrite.

A clear fix that does not join

The rule, checked against rewrite, the mode run in full:

THE RULE	AS TYPED	REWRITE	
Search clearly above likely ranges must not overlap	66-75%	69-78%	✗
Right answers not lower	280	296	✓

The fix that worked: resolve follow-ups only

It found 28 passages that typed questions missed and lost just 1. It calls the model for only the 40 questions with earlier turns, and leaves codes alone.

The lesson: a real fix can hide in the total

40 follow-ups are a tenth of the 400. Fixing nearly all of them still cannot lift the overall number clear of its likely range. Judge a targeted fix on the questions it targets, and track those by type.

What this does not prove

- **One rewriting model, one instruction per mode.** A better-worded instruction could keep codes intact.
- **Small samples:** 20 two-part and 20 bridge questions.
- **The dev choice was close:** made on 20 hard questions, with the three modes within two of each other.
- **No search, read, search again.** Bridges need it; that is Season 5.
- **The judge** is gemini-3.6-flash, checked against E3's sample.

Run it yourself

All three modes, both rounds. The rewritten queries are in each run's log.

```
# rebuild every result and chart: no key, no cost
$ uv run lab rag analyse episodes/s2-e6-rewriting/config.yaml

# round 1 on dev, then the full run (keys needed)
$ uv run lab rag run episodes/s2-e6-rewriting/config.dev.yaml
$ uv run lab rag run episodes/s2-e6-rewriting/config.yaml
```

Cost	\$1.33 for the whole episode, almost all of it judging the full run.
Instructions	src/lab/rag/prompts/rewrite_*.md, one per mode.

Real users ask in conversations

A chat assistant that searches only the last message fails every "what about...?" Here, one call costing under a hundredth of a cent took follow-ups from 7 found in 40 to 34.

Ask your team

When a user asks a follow-up, what does our search actually receive?

Ask your team

If we rewrite questions, which kinds of question did it make worse?

Ask your team

Which of our questions need two lookups in a row? Those need more than a rewrite.

Stage 4 tested. Not joined.

1 Load	2 Chunk	3 Embed	4 Rewrite	5 Search	6 Filter	7 Rerank	8 Prompt	9 Answer	10 Cite	11 Score
-----------	------------	------------	--------------	-------------	-------------	-------------	-------------	-------------	------------	-------------

	SEARCH	RIGHT	NOT FOUND	LEAKS
After E5 as typed	281 of 400	280 of 400	42 of 120	76 of 480
Rewrite tested, not joined	296 of 400	296 of 400	43 of 120	98 of 480
After E6 unchanged	281 of 400	280 of 400	42 of 120	76 of 480

Lime: joined. Coral: tested, not joined. The bottom row carries forward to E7.

Decision rule

Rewrite a question only where tests show that kind of question fails. Rewrite follow-ups, not everything, and weigh the extra call against what it recovers.

Next: S2 E7 Reranking

Of the 120 wrong answers, 89 had the right passage in search's top 50, just not in the top 5. A second, slower look reorders them.

Where this fits

The series map

S1 LLMs	S2 Retrieval	S3 Strategy
S4 Adapting	S5 Agents	S6 Economics
S7 Data	S8 Evaluation	S9 AI teams
S10 Reliability	S11 Security	S12 Delivery
S13 Performance	S14 Resilience	S15 Governance

☐ Leadership season

 **Follow**

linkedin.com/in/brendan-kelly-irl

 **Episode guide**

brendankellyai.github.io

 **Code**

github.com/BrendanKellyAI/applied-ai-lab