

Season 2: Retrieval and RAG

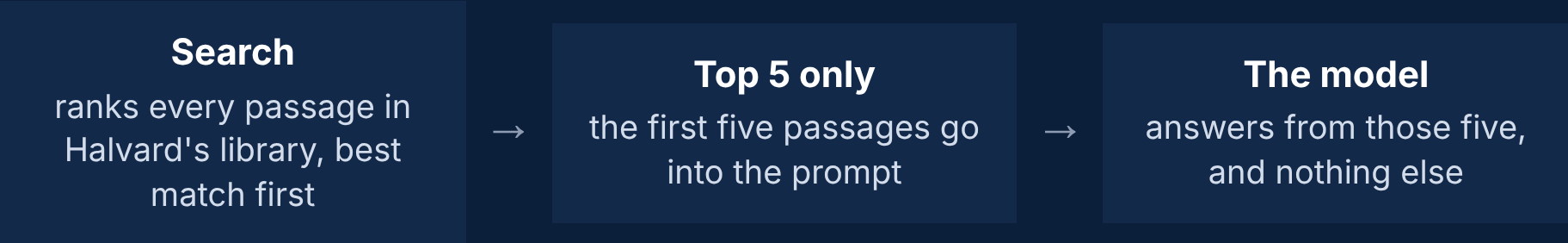
Reranking.

Search often finds the right passage, then ranks it too low for the model to see. A second, closer read fixes that. Tested against the 520 test questions.

The answer was there. The model never saw it.

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

HOW OUR SYSTEM ANSWERS A QUESTION, SO FAR



A REAL TEST QUESTION

What does error E-2194 mean? Search ranked the passage with the answer 10th

Search **found** the right passage, but put it 10th. The model only saw the top 5, so it could not answer. Across all 400 answerable questions, this is where the right passage landed:

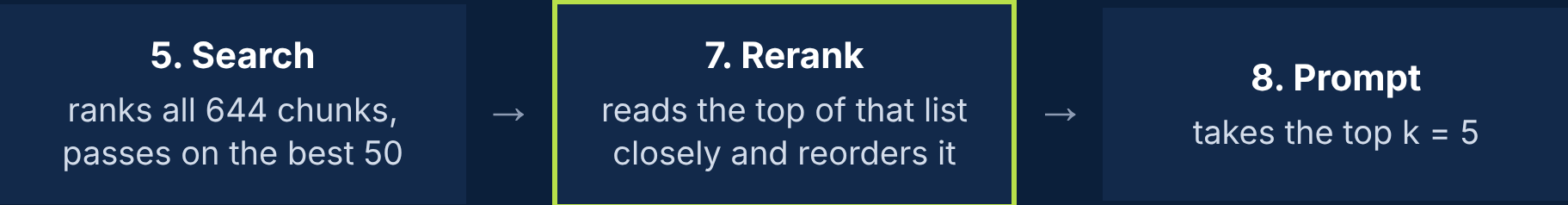


This episode

58 right passages sat just below the line. Can a second, closer read of search's top results move them up into the top 5?

Where stage 7 sits

Stage 7 runs live, on every question, between search and the prompt.



Until now	Stage 7 passed the list straight through. Search's order was the final order.
What changes	Only the order of the top of the list. Nothing is added or removed.
The shortlist	How many of search's top passages the reranker reads. It can only promote what is on it.

Why two steps?

Search must be fast enough to score every chunk in the library. A reranker can afford to be slower and more careful, because it only reads a handful. Next slide: why that makes it more accurate.

Apart, or together

The difference between search and a reranker is when the question meets the passage.

SEARCH (SINCE E2)

Read apart

Question

↓

[0.01, ...]

Passage

↓ ahead of time

[0.02, ...]

↓ compare two lists of numbers

a score, in microseconds

Each passage is turned into numbers **without ever seeing the question**. Fast enough for 644 chunks, but coarse.

RERANKER

Read together

Question + Passage

↓ one model reads both at once

does this passage answer this question?

↓

a sharper score, in seconds

The model sees the question's words **beside** the passage's, so it can check the exact code, product and fact. Too slow for 644; fine for 20.

So the pattern is
Search wide and fast for a shortlist. Then read the shortlist closely.

Brendan Kelly

4 / 21

Two kinds of reranker

Cross-encoder the classic choice

A small model trained for one job: read a question and one passage together, return a score. Run it once per passage on the shortlist, then sort. Here: BAAI's bge-reranker, run on our own machine.

LLM reranker this season's choice

A general model reads the question and the whole numbered shortlist **in one call**, and replies with the passage numbers, best first. Here: gpt-6-luna, the cheap model already answering.

THE LLM RERANKER'S WHOLE INSTRUCTION

"You rank passages from Halvard Home's support articles and policies by how well each one answers a customer's question. **The best passage states the exact fact asked for, about the exact product, code or region named.** Reply with the passage numbers only, best first, separated by commas, and include every passage."

Why the season uses the LLM

Both were tried. On the 40 questions both ran on, they were equally accurate. The difference was time.

CROSS-ENCODER, ON A LAPTOP CPU

62 s

per question, for a shortlist of just 10. Passages here are about 840 tokens, and each one is a separate pass. All 520 questions with a shortlist of 20: about **16 hours**.

LLM RERANKER, ONE CALL

~1 s

per question, for a shortlist of 20. About 10,300 tokens in, 43 out: roughly **\$0.001** a question.

The lesson: it depends where you run it

On a GPU, a cross-encoder is many times faster and costs nothing per question. On this machine, the "free" local model was the slow and expensive one. Measure on your own hardware.

Stage 7: Rerank

WHAT CHANGES IN THIS STAGE

The top 20 of search's list go to the LLM in one call. Its order replaces search's order for those 20. Anything it leaves out keeps its search place, after the ones it named. Places 21 to 50 never move.

```
E6 rerank: {mode: passthrough}
```

```
E7 rerank: {mode: llm, shortlist: 20, model: gpt-6-luna}
```

```
src/lab/rag/rerank.py
```

```
the whole function
```

```
def rerank(ranked, scores, shortlist):  
    """Reorder the top `shortlist` of the ranked passages by  
    the reranker's score for each, best first. The rest keep  
    their search order."""  
    top = sorted(ranked[:shortlist], key=lambda p: -scores[p])  
    return top + ranked[shortlist:]
```

shortlist: 20

How far down search's list the reranker reads. A passage search ranked 21st can never reach the prompt.

E-2194, three episodes later

THE CODE QUESTION FROM E5

What does error E-2194 mean? Answer: an interrupted factory reset on the Ashby tumble dryer

E5	Vector search ranks it behind other "error E-something" articles.	10th
E6	Rewriting adds generic words, and pushes it further down.	17th
E7	The reranker reads the 20 passages beside the question, sees "E-2194" in exactly one of them, and puts it first.	1st

Wrong answer, now right

Search had already found it: 10th is inside the shortlist of 20. It only needed something that reads closely enough to see the exact code. Of the 40 code questions, found went from 15 to 26.

Our Extended Care question was already 2nd. The reranker kept it 2nd: nothing to fix.

The test: three versions of stage 7

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

Search order

the reference

No reranking. Today's system.

LLM, top 10

The reranker reads search's top 10.

LLM, top 20

The reranker reads search's top 20.

1 Round 1: search only, all three

All 400 answerable questions. Checked by code: is the right passage in the top 5 after reranking?

2 Round 2: the full system, one shortlist

Chosen on the dev questions: top 20 found 56 of 70, top 10 found 52. Top 20 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.

Four kinds of question to watch

Halvard's test questions come in kinds, 20 or 40 of each. These four are where search struggled most, so they show what reranking changes.

Codes

40 questions

Ask about one error code or part number. The look-alike codes in other articles make them hard for search.

"What does error E-2194 mean?"

Two-part

20 questions

Ask two things at once, answered in two different passages. Only a hit if **both** are in the top 5.

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

Bridge

20 questions

Need one fact to find another: first which region a postcode is in, then that region's time.

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

Versions

40 questions

The library keeps an old and a current copy of some policies. The answer must come from the **current** one.

"How long should the Tessa dishwasher cool before I clean it?"

It can only promote what it is shown

Search: right passage in the top 5, after reranking, for each of the four kinds from the last slide.

	CODES OF 40	TWO-PART OF 20	BRIDGE OF 20	VERSIONS OF 40	ALL 400
Search order the reference	15	0	0	36	281
LLM, top 10	20	3	1	39	298
LLM, top 20	26	8	5	40	316

Read across	Every kind rose with reranking, and rose further with the longer shortlist. Codes: 15, then 20, then 26.
Why 20 beats 10	From slide 2: 28 right passages sat at 6th to 10th, but 58 at 6th to 20th. A longer shortlist gives the reranker more to rescue.
Nothing else fell	The other kinds held. Only follow-ups got worse: slide 13.

Reading together helps a bridge

THE BRIDGE QUESTION FROM E6

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

Passage 1: which region?

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

+

Passage 2: that region's time

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

E6: a rewrite could not bridge, because it never sees a passage. Search had both passages in its top 20, the lower one at 18th. The reranker reads them side by side, can link "LL" to "Wales and West", and lifted both into the top 5. The lower one now sits **4th**.

5 of 20, from 0

Not a real fix: both passages must already be on the shortlist. A bridge still needs search, read, search again (Season 5). But reading candidates together recovers some.

What a reranker cannot fix

Never found

11 answers were not in search's top 50, and 50 more sat at 21st to 50th, beyond the shortlist. The reranker never saw them.

Follow-ups

7 found, then 5. "What about the Mira robot vacuum?" is just as vague to the reranker as to search. E6's rewrite is the fix for these.

A few losses

3 passages that were in the top 5 were pushed out, to 6th or 11th. Two of the three were follow-ups.

The lesson

Each stage fixes its own kind of failure. Reranking fixes "found but ranked too low". It does not fix "never found" (search, E5) or "the question is unclear" (rewriting, E6).

Better search, better answers

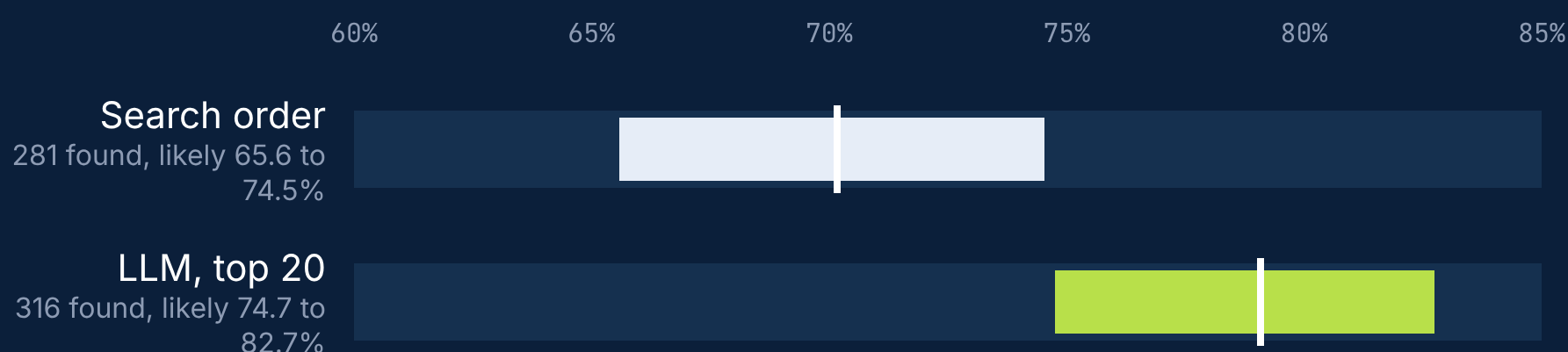
Round 2: LLM top 20 against search order, the whole system, all 520 test questions.

	SEARCH	RIGHT	NOT FOUND	LEAKS
Search order the reference	281 of 400	280 of 400	42 of 120	76 of 480
LLM, top 20	316 +35	316 +36	41 -1	64 -12

Question by question	Of the 400, the reranker got 43 right that search order got wrong, and 7 the other way.
Ranked too low	Wrong answers caused by a passage ranked below 5th: 89 before, 55 after.
Leaks	Down to 64, but still far from zero. That is E8's job.

It joins, with almost nothing to spare

The rule, set in advance: a new stage joins only if its likely search range sits wholly above the old one. No overlap at all.



The bars miss each other by **0.2 points**. Had the reranker found **one** fewer passage, 315 instead of 316, they would touch, and by the rule it would have stayed out.

So is the gain shaky? No

The no-overlap rule is deliberately strict. The sharper check from E5, question by question, is clear: the reranker got **43** right that search order got wrong, and only **7** the other way. That is not luck.

The lesson

A strict rule protects you from fooling yourself, but it can only just pass a real gain. Report how close the call was, and use the question-by-question check to see how strong the gain really is.

What this does not prove

- **The cross-encoder at scale.** Tested on 40 questions, on a CPU. On a GPU the time comparison could reverse.
- **One LLM, one instruction.** A small model ranking a list can drop or repeat passages.
- **A prediction with no room.** We expected fewer look-alikes ranked above the answer. But since E4's section chunks, none were. Zero cannot fall. The claim was written from E2's numbers and not rechecked.
- **The judge** is gemini-3.6-flash, checked against E3's sample.

Run it yourself

Both shortlists, the full run, and the cross-encoder comparison.

```
# rebuild every result and chart: no key, no cost
$ uv run lab rag analyse episodes/s2-e7-reranking/config.yaml

# the full run (keys needed)
$ uv run lab rag run episodes/s2-e7-reranking/config.yaml

# the local cross-encoder, 40 questions
$ uv run --extra rerank python episodes/s2-e7-reranking/cross_encoder_subset.py
```

Cost

\$1.89 for the episode: \$0.95 of reranking calls, \$0.94 of judging.

Find out where the answer sits

Here, one call costing a tenth of a cent per question added 36 right answers out of 400. It worked because search was already finding the answers, just ranking them too low.

Ask your team

When we get an answer wrong, how far down search's list was the right passage?

Ask your team

What does a reranker cost us per question, in money and in waiting time, on the hardware we actually run?

Ask your team

How long is our shortlist, and how many answers sit just below it?

Stage 7 changed. It 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 E6 search order	281 of 400	280 of 400	42 of 120	76 of 480
After E7 LLM rerank, top 20	316 of 400	316 of 400	41 of 120	64 of 480

Lime: joined. Coral: tested, not joined. Every later episode starts from the bottom row.

Decision rule

Add a reranker when the right passage is found but ranked too low. Give it a long enough shortlist, and measure what it costs per question.

Next: S2 E8 Filters

64 times in 480, a public customer's prompt still held staff-only text. Search should never have offered it.

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