

Season 2: Retrieval and RAG

Why retrieval exists.

What it fixes, what it does not, and the new failure it introduces.

The answer is in one sentence no model has read

A HALVARD HOME CUSTOMER ASKS

How much longer does Extended Care keep my warranty going?

The answer, 58 months, sits in one sentence of Halvard's 27,700-token warranty policy. No model was trained on it. So a model has three options:

Option 1 Guess	FLUENT, PLAUSIBLE, WRONG Fill the gap with a figure typical of other companies' plans.
Option 2 Decline	HONEST, AND NO HELP Say it does not know Halvard's terms.
Option 3 Read it	RETRIEVAL Be handed the right sentence at the moment of the question, and answer from it.

Your data is not in the model

Four reasons, each one measured or shown in Season 1.

Training stops at a date

A policy changed last month, a product launched last week: the model cannot know either.

Training cut-off

Your documents were never in it

Contracts, tickets, manuals and internal wikis are not public training data.

Halvard: *invented*, *unseen*

Every call starts fresh

The model holds only what you send it in this call. It remembers nothing between calls.

S1 E1

Gaps get filled

Asked about "the Data Protection Act 2018" without saying Irish, all six answers described the UK Act.

S1 E9

So

For anything specific to your organisation, the model has to be told. Retrieval is how it gets told, on every question.

Closed book against open book

RAG, retrieval-augmented generation, turns a closed-book exam into an open-book one. Search decides which pages are open.

WITHOUT RETRIEVAL

Closed book

Question



Model's memory



Answer

- Knows only its training data
- Cannot say where an answer came from
- Fills gaps with what is likely

WITH RETRIEVAL

Open book

Question



Search your documents



Top passages + question



Answer, with citations

- Answers from your pages, today's version
- Can point to the passage it used
- Only as good as the pages it is handed

One question, followed through

The season's final system, on the Extended Care question.

1

THE QUESTION

How much longer does Extended Care keep my warranty going?

2

SEARCH

Compares the question with **every passage** in Halvard's library, keeps the closest, and reranks the best 20.

3

TOP RESULT

Warranty > Extended Care > What the plan adds

"Extended Care extends the standard warranty by **58 months**. It adds protection against covered mechanical and electrical faults..."

4

THE PROMPT

The top **5 passages**, numbered P1 to P5, plus the question and an instruction: answer only from these, and cite them.

5

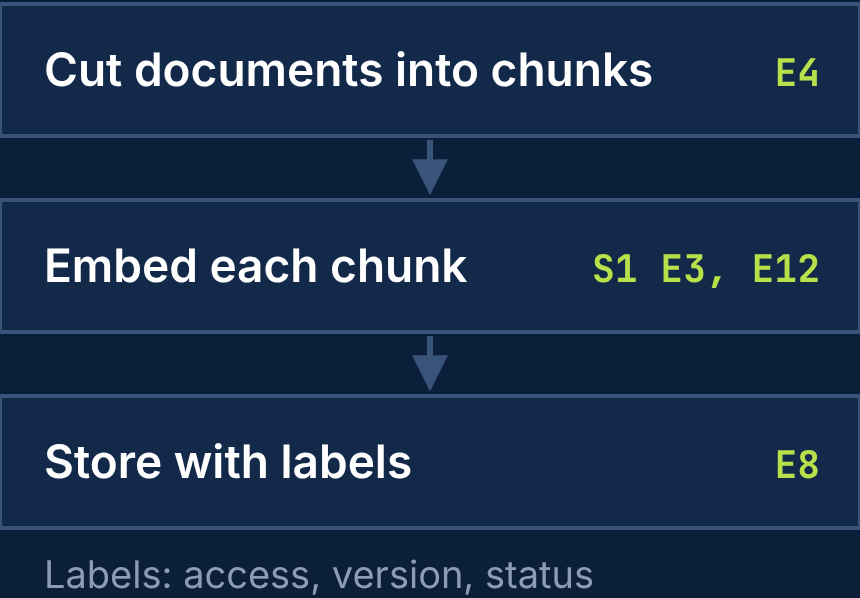
THE ANSWER

"Extended Care extends the standard warranty by 58 months. [P1]"

Two pipelines, one decision

One prepares the library when documents change. The other runs on every question. Each box is an episode this season.

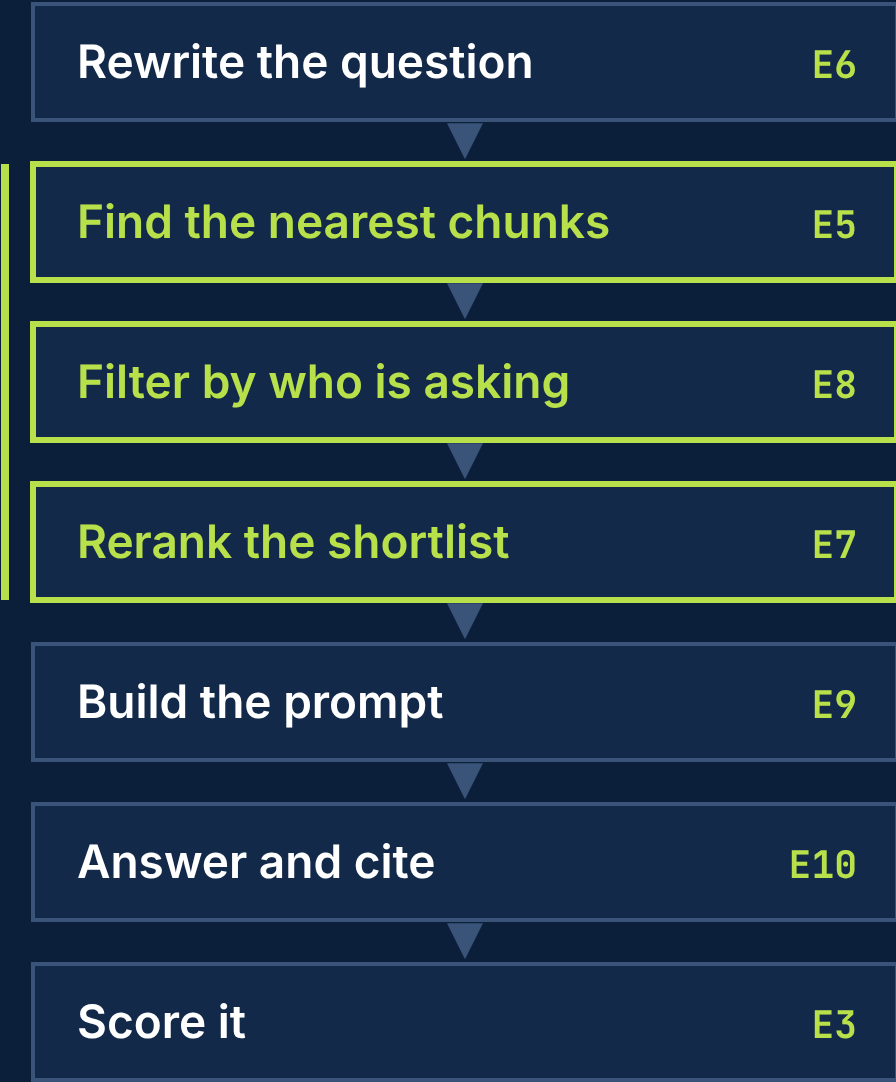
Once, when documents change



The lime steps decide what the model can see.

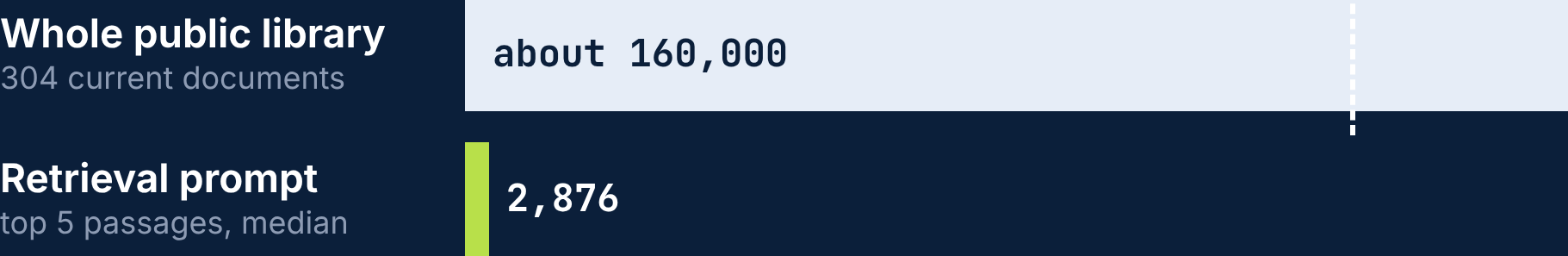
Questions and chunks go through the same embedding model, or their positions cannot be compared (E12).

On every question



Why not send the whole library?

Tokens the model reads for one question, measured on Halvard.



Dashed line: a 128,000-token context window. The library does not fit.

It can work	S1 E7 found a single fact in 270 of 270 calls, up to 64,000 tokens. Size alone is not the problem.
It does not fit	Halvard's public library is bigger than the window. Most real libraries are far bigger.
Paid every call	Every token sent is billed on every question, about 55 times the retrieval prompt here.
Untested shapes	Look-alike passages and two-part questions were not tested in S1 E7. E13 runs them head to head.

What retrieval fixes

Stale facts

Update the index, not the model. A new policy can be answered from the moment it is indexed.

E8: `versions`

Private data

Answers can draw on documents the model never saw, like Halvard's warranty policy.

E2: `the first system`

Traceability

Each answer can point to the passage it came from, and code can check the pointer.

E10: `citations`

Scope

Search only what this asker may read: public documents for customers, internal ones for staff.

E8: `permissions`

Read this with the next slide

Each fix depends on search returning the right passage. That step has its own failure.

Close is not correct

Search ranks passages by closeness in meaning. Closeness, scored by an embedding model in S1 E3:

OPPOSITE "Shares rose 5% today." against "Shares fell 5% today."



OPPOSITE "The drug is safe." against "The drug is not safe."



SAME MEANING "The drug is safe." against "The medicine carries no risk."



Cosine similarity, text-embedding-3-small. 1.0 would be identical in meaning to the model.

On Halvard

To an embedding, error E-2194 looks a lot like E-1294 and E-4194, which sit in other articles. In the first system, the right article for "What does error E-2194 mean?" ranked 11th, out of the prompt.

A close match is a candidate, not an answer. Later stages exist to sort candidates (E5, E7).

Retrieval moves the failure. It does not remove it.

WITHOUT RETRIEVAL

Wrong about the world

The model guesses a fact it was never given. Nothing in the answer says so.

WITH RETRIEVAL

Wrong about the source

The model misstates a passage it was given, or answers from the wrong passage.

The new failure is worse in one way: **an unfaithful answer arrives with a citation attached**, so it looks more trustworthy than a plain guess.

35

of 400 citations

In E10, the final system's citations were checked in code. **35 of 400 pointed to a passage that did not hold the evidence.** Most sat beside a correct citation; 5 answers cited nothing else.

What retrieval does not fix, and where this season tackles it

FAILURE	WHAT IT LOOKS LIKE	EPISODE
Missed passage	The right passage is never found, so the model answers without it.	E4 E5 E6
Ranked too low	Found, but below the cut-off, so it never reaches the prompt.	E7
Bad source	A superseded policy or a withdrawn notice is retrieved as readily as a current one.	E8
Leaked passage	A staff-only document reaches a customer's prompt.	E8
Misreading	The right passage is in the prompt; the answer still gets it wrong.	E9 E10
Two-step questions	Retrieval supplies facts, not the steps that combine them (S1 E10).	E6 E13
Its own score	Quality is usually scored by another model, and S1 E9 showed a score can be wrong.	E3

Design for it from day one

Log retrieval

Store the passages behind every answer, so a wrong answer can be traced to its cause (E14).

Allow "not found"

Make it a valid answer, and test it with questions your documents cannot answer (E9, E10).

Score apart

Measure whether search found the passage separately from whether the answer used it (E3).

Filter first

Apply permissions and versions before ranking, never after (E8).

Pin the model

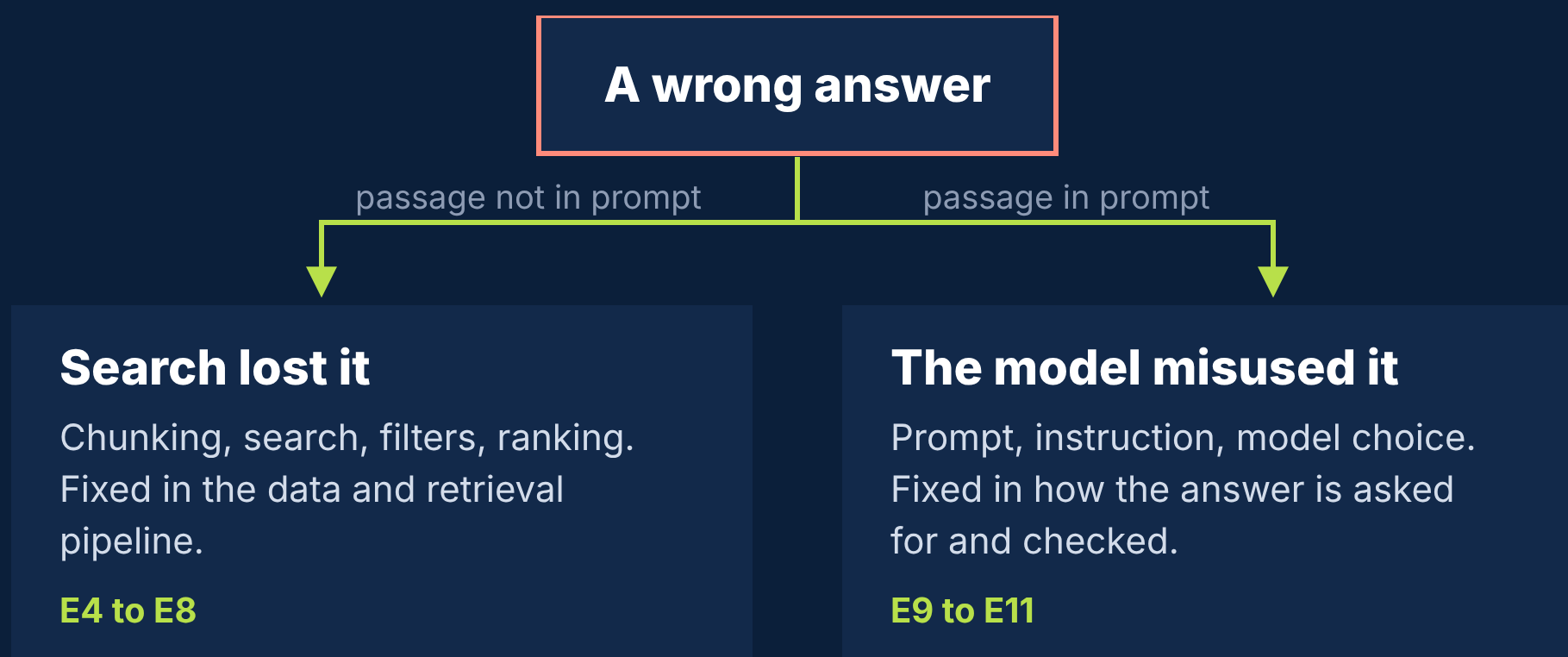
One embedding model for documents and questions, versioned. A swap means re-embedding everything (E12).

Start simple

If the documents fit the window and rarely change, try sending them before building a pipeline (E13).

A wrong answer has two possible owners

When a grounded assistant is wrong, either search lost the right passage or the model misused it. The fixes differ, and so do the people who make them.



Ask your team

When our assistant is wrong, can we tell whether search or the model was at fault, and how often each happens?

What the season does with this

The same 520 Halvard questions, through the first system (E2) and the final one. Each step in between is an episode.

Search	Right answers	Leaks
right passage in top 5, of 400	correct and faithful, of 400	staff passage to a customer, of 480
First239	First217	First51
Final344	Final354	Final0

Almost all of the gain came from the steps that decide what the model sees: chunking (E4), reranking (E7) and filters (E8). E16 tells the full story.

Decision rule

Use retrieval when the answer lives in your own data. Then log what it retrieved, and measure search and answer separately.

Next: S2 E2 Your first RAG system

Every stage built with the simplest choice, run on all 520 questions, and measured.

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