Session 6.2

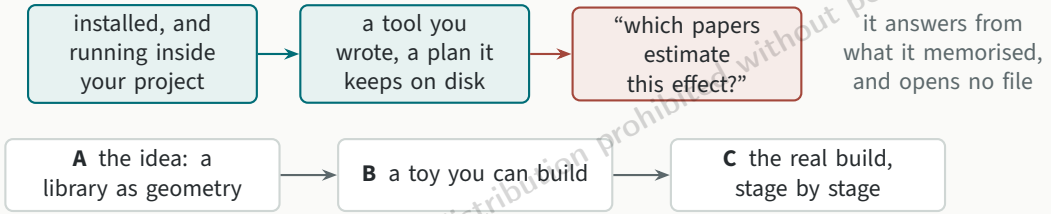
Build Your Own RAG: Give Your Agent a Library



Zhigang Feng

© Zhigang Feng
Redistribution prohibited without permission

Your Agent Runs Your Code, and Cannot Read Your Papers



installed, and running inside your project → a tool you wrote, a plan it keeps on disk → "which papers estimate this effect?" — it answers from what it memorised, and opens no file

A the idea: a library as geometry → **B** a toy you can build → **C** the real build, stage by stage

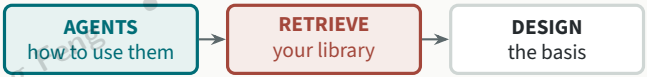
© Zhigang Feng — Redistribution prohibited without permission

- An agent now runs commands, calls tools and keeps a plan on disk. The papers on your own disk are still outside the loop.
- Ask it which papers estimate your effect: it answers from what it memorised, and opens no file.
- **This deck puts your library inside the loop: the idea, a toy, then the real build.**

Session 6.2

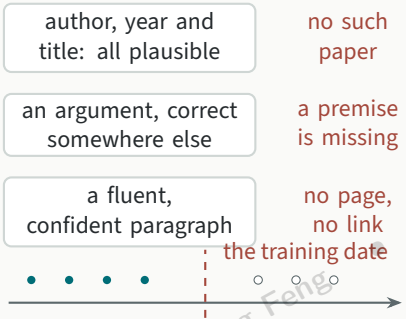
The Idea

Why the model cannot just answer, and the one move that fixes it



© Zhigang Peng Redistribution prohibited without permission

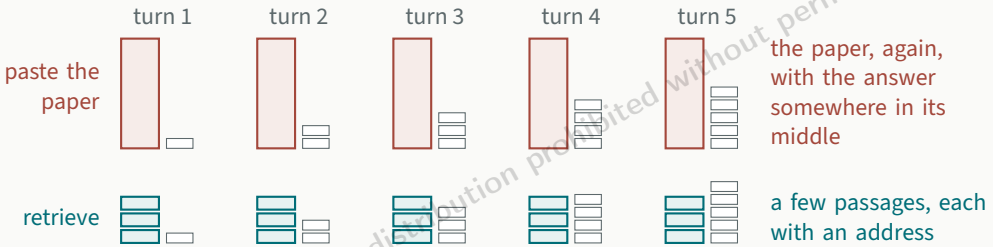
Three Symptoms, One Cause



- A fabricated citation, a method applied where it does not hold, a fluent paragraph with no page: three faces of one problem.
- The model answers from parameters frozen at a training date. Your papers were never in them, and nothing it says points to a page.
- **It does not have your literature, and what it has it cannot cite.**

© Zhaogang Feng · Redistribution prohibited without permission

Pasting the Paper Is Not the Fix

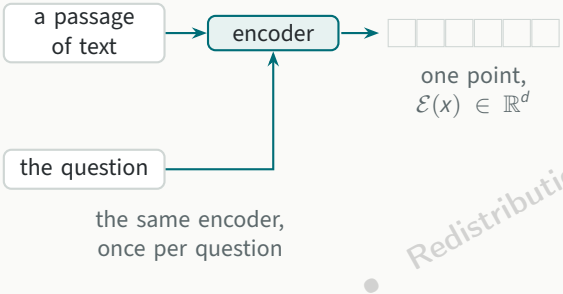


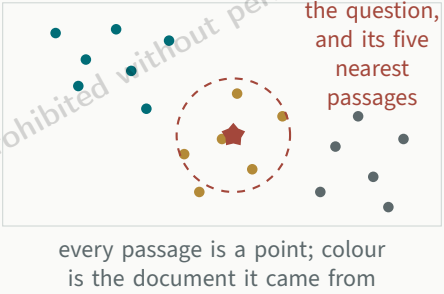
© Shuguang Sun · Redistribution prohibited without permission

- Paste a paper and every later question re-sends it; the passage you need sits in the middle, where a long context is used least reliably.
- Retrieval sends only the few passages the question needs, each with its address.
- **Send the evidence, not the library.**

Source: Liu et al. (2024), "Lost in the Middle", *TACL* 12, 157–173.

The Key Idea: Put the Library in a Space Where Nearness Means Meaning

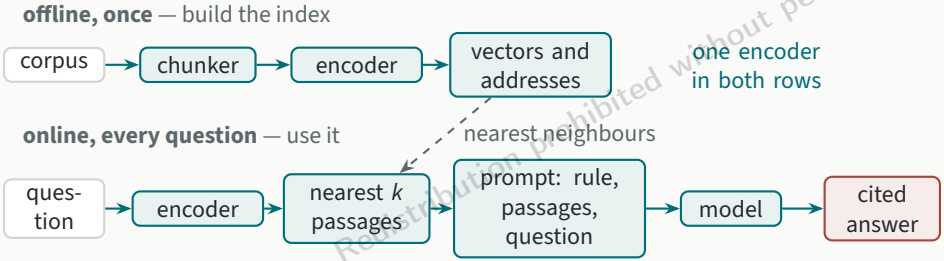




Redistribution prohibited without permission
© Zhiwei Feng

- Every passage is turned into a point by an encoder, once. Passages that say similar things land near each other: the geometry of word vectors, now for whole passages.
- A question is turned into a point by the same encoder. Its evidence is the nearest points, read back as text with their addresses.
- **The library becomes geometry, and search becomes nearest neighbours.**

Build Once, Ask Many Times

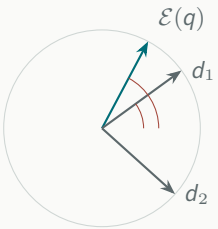


© Liang Feng Redistribution prohibited without permission

- Offline, once: cut, encode, store the vectors beside their addresses.
- Online, per question: encode it, take the nearest passages, give the model one rule: answer from these, cite each.
- **Both rows use one encoder: similarity exists only inside one space.**

Source: Lewis et al. (2020), *NeurIPS* 33, 9459–9474.

Why Cosine: Compare Directions, Not Lengths



cosine compares angles, so passage length drops out

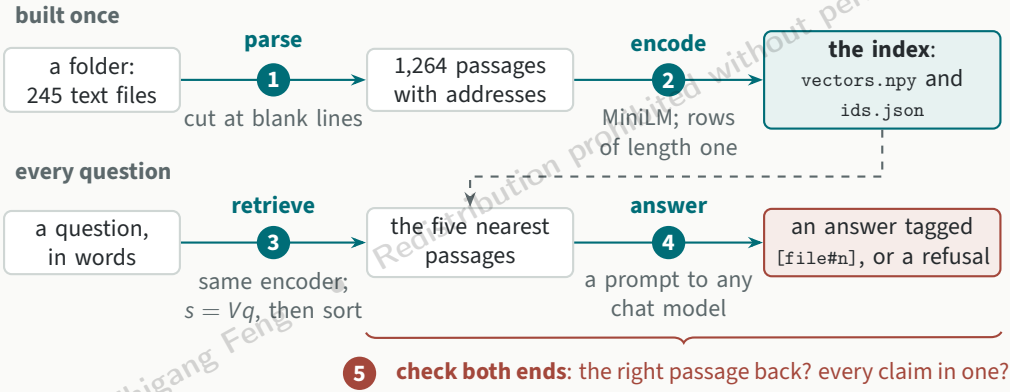
- Score a passage by the angle between its vector and the question's:

$$\mathrm{sim}(q, d_i) = \frac{\mathcal{E}(q)^\top \mathcal{E}(d_i)}{\|\mathcal{E}(q)\| \, \|\mathcal{E}(d_i)\|},$$

  and with unit-length vectors it is one dot product.
- **Top-k**: keep the k nearest. A **vector store** is the table of vectors and addresses that answers this quickly.
- **Nothing here understands the question; it measures angles.**

© Zhigang Feng
Redistribution prohibited without permission

A Toy You Can Build: Five Steps From a Folder to a Cited Answer



© Zhigang Feng Redistribution prohibited without permission

- Steps 1–2 run once, Steps 3–4 per question; Step 5's questions come first.
- **Every box is a thing you can open; each numbered arrow is one short listing.**

What the Toy Needs, Built Once: Parsing, Encoding, Storing

step	what you need	in the toy	it breaks when
1 parse	plain text saved as UTF-8; a rule that cuts it; an address on every piece	cut at blank lines; pieces over 40 characters kept; median 366 characters	the files are PDFs or scans: convert first
2 encode	`sentence-transformers`; one encoder, fetched once, then read from disk	all-MiniLM-L6-v2, a small BERT (6 layers), 91 MB; 384 numbers of length one; reads 256 word pieces, so 11 passages are cut	the question is read by another encoder
store	NumPy; the matrix and its addresses, in one order	`vectors.npy`: 1,264 × 384 numbers × 4 bytes = 1.9 MB; `ids.json`	one file is rebuilt, the other is not

- One install: `pip install sentence-transformers`; NumPy and PyTorch are in the course environment.
- **No server, no database: a laptop builds the index once, as two files.**

Source: The toy's run on 20 October 2026; the encoder's numbers from its own files.

© Zhigang Feng — Redistribution prohibited without permission

What the Toy Needs, Per Question: Retrieval, a Prompt, a Check

step	what you need	in the toy	it breaks when
3 retrieve	the same encoder; one product, one sort; a filter on the address for dates and names	$s = Vq$: 1,264 cosines from one product; the top $k = 5$ kept	the question names a date: right topic, wrong year
4 answer	a prompt (the rule, each passage with its address, the question) and any chat model	2,518 characters for the toy's question; the lab prints it and sends nothing	the rule or the addresses are left out
5 check	questions with known answers, and some the corpus cannot answer, written first	the lab's 26: 20 answerable, 6 not; the right passage back? each claim in one?	only the end that looks good is tested

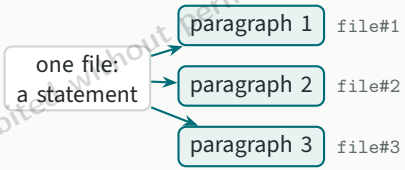
© Shuang Feng · Redistribution prohibited without permission

- Step 4 is the only step that needs a chat model: an API, a local model, or the agent itself.
- **Everything before the answer runs offline and gives the same rows on every run.**

Source: The toy's run on 20 October 2026; the lab's graded questions.

Step 1: A Folder of Text Becomes a List of Passages

```
from pathlib import Path
import json, numpy as np
docs, ids = [], []   # passage, address
for f in sorted(Path("corpus").glob("*.txt")):
    paras = f.read_text("utf-8").split("\n\n")
    for n, p in enumerate(paras):
        if len(p) > 40 and p[0] != "#":
            docs.append(p)
            ids.append(f"{f.stem}#{n}")
```



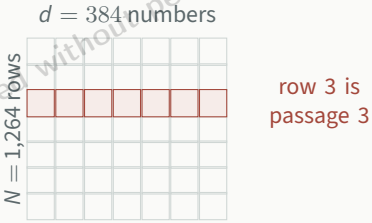
245 statements,
1994-02-04 to 2026-06-17,
become 1,264 passages

© Zhaoyang Feng · Redistribution prohibited without permission

- The corpus is a folder. Each file is read as text and cut at paragraph breaks into passages of a few hundred characters.
- Each passage keeps its address, file and position. That address is what a citation will print.
- **The unit you cut is the unit you will cite.**

Step 2: Encode Every Passage Once, Keep the Matrix

```
import sentence_transformers as st
name = "all-MiniLM-L6-v2"
enc = st.SentenceTransformer(name)
V = enc.encode(docs, normalize_embeddings=True)
# V: N x 384, one row per passage, length one
np.save("vectors.npy", V)
json.dump(ids, open("ids.json", "w"))
```



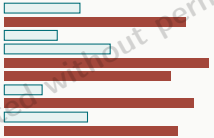
`vectors.npy` and `ids.json`, in the same order

- One call turns N passages into an $N \times d$ matrix, one row per passage; rows of length one, so cosine is a dot product.
- Save the matrix and the address list side by side. That pair is the whole index.
- **This is the expensive step, and it runs once.**

© Zhigang Feng • Redistribution prohibited without permission

Step 3: Encode the Question, Take the Nearest Rows

```
q = enc.encode([question],
               normalize_embeddings=True)[0]
s = V @ q                  # cosine, all rows
top = np.argsort(-s)[:5]   # the five nearest
for i in top:
    print(ids[i], s[i].round(2), docs[i][:60])
```



s: one score per row; red, the five kept

N scores from one product, sorted once

printed, for *question* = "What did the Committee say about inflation in March 2022?"

`fomc_20090128#3`	0.66	In light of the declines in the prices of energy and other c…
`fomc_20180131#2`	0.66	Consistent with its statutory mandate, the Committee seeks t…
`fomc_20131218#2`	0.65	Consistent with its statutory mandate, the Committee seeks t…

- Right topic, wrong year: the encoder reads meaning, and the date lives in the address.
- **Search is one line of algebra; the date needs a filter on the address.**

Source: The listing on this page, run on the session's 245 statements on 20 October 2026; the three best of its five rows.

© Zhigang Feng Redistribution prohibited without permission

Step 4: Build the Prompt; Answer With Citations, or Decline

```
context = "\n\n".join(f"[{ids[i]}] {docs[i]}"
                      for i in top)
prompt = f"""Answer only from these passages.
Tag every claim with its [address].
If the answer is not in them, say so.

{context}

Question: {question}"""
answer = model(prompt)    # any chat model
```

the rule: only from these; tag each claim; if absent, say so

the passages: the five nearest, with addresses

the question

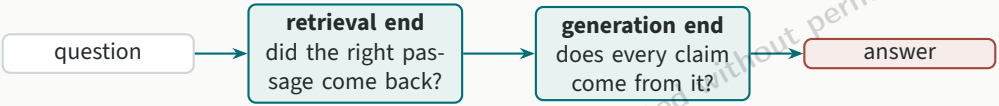
the answer: clauses tagged `[file#n]`, or "not in the passages"

Redistribution prohibited without permission

© Eugene Yan

- Three parts: the rule, the passages with their addresses, the question. Given the rows of Step 3, the right answer is a refusal: the model fills no gap from memory.
- The only place a generative model runs. Everything before it exists to put good evidence in front of it.
- **"Not in the passages" is the system working; a fabrication costs the paper.**

Step 5: Check Both Ends Before You Trust It



a good score at one end and a bad one at the other is not a good system: it is a system that fails in a way you have not measured

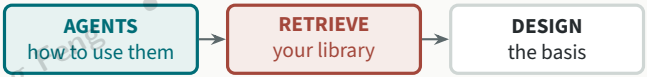
© Zubair Feroz · Redistribution prohibited without permission

- Before you build, write questions with known answers, and some the corpus cannot answer.
- Retrieval end: did the right passage come back, recall at k? Generation end: does every claim in the answer appear in the passages, **faithfulness**? A refusal rule is a threshold on the best score.
- **Both ends must pass. They fail for different reasons and are repaired in different places.**

Session 6.2

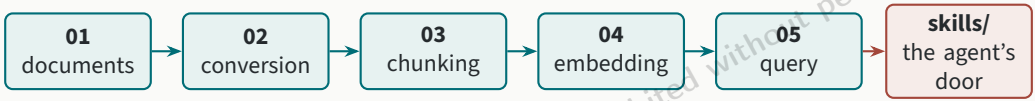
The Real Build, Stage by Stage

The toy's five steps, each now a folder, a tool and a check



© Zhigang Feng Redistribution prohibited without permission

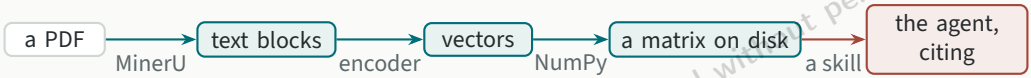
The Build Is Five Numbered Folders



© Zhigang Feng Redistribution prohibited without permission

- A stage reads the folder before it, writes its own; a failure is fixed in one place.
- The toy's five steps, now five folders: what changes is scale, formats, checks.
- **Number the stages, and never let one write into another.**

The Four Tools, in Plain Words, and Where You Met Each

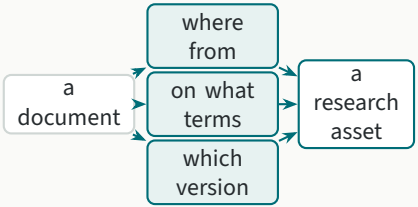


the tool	how it works	you met it as
MinerU, a layout model	reads a page as a picture and writes out its blocks, each with its type and page	5.1: a PDF is a picture, not tokens; 5.3: read, write
the encoder, BGE-M3, open weights	reads a passage both ways into one unit vector; two languages, one space	5.2: word vectors, cosine; 5.3: the pinned version
a NumPy store, the matrix on disk	one row per chunk; search is one product Vq and a sort, a query against its keys	Sessions 2–4: matrix products; 5.2: query–key scores
a skill: a manual and a script	read only when a question needs the library: when to run the script, how to cite	6.1: a folder opened when needed; the loop with a tool

- Retrieval is attention over a library: the question is the query, the vectors are its keys.
- **No magic: a reader for pictures, an encoder for meaning, a matrix, a manual.**

© Zhigang Feng. Reproduction prohibited without permission

Stage 1, Documents: The Corpus Is a Choice, and the Folder Is Its Metadata

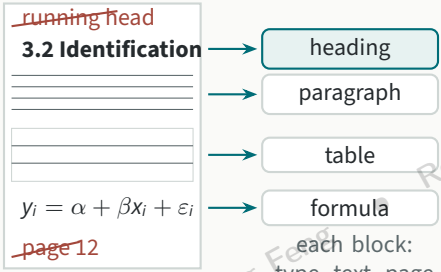


- What is left out cannot be retrieved. No licence, no provenance, no frozen version: a pile of files, not a corpus a referee can re-run.
- The path is the topic, and a manifest row carries each file's content hash, so a rerun passes over what is already converted.
- **The cheapest metadata is the kind you cannot forget to update.**

© Zhaoran Feng · Redistribution prohibited without permission

Stage 2, Conversion: A PDF Becomes Blocks of Text, Each With Its Page

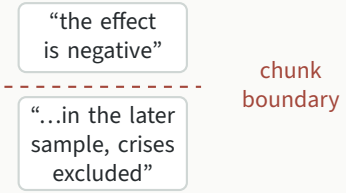
```
mineru -p staged/ -o markdown/ -b vlm-auto-engine  # one engine start per batch
pandoc paper.tex -t gfm -o paper.md                # TeX, HTML, EPUB: no model
```



- A PDF is a picture. A layout model on a GPU reads it into blocks, heading, paragraph, table, formula, each with its page; running heads and page numbers are dropped.
- The engine loads once, then every page is cheap: stage the batch, start it once. Other formats need no model.
- **Normalise early: every format becomes one block record, and the page travels with the text.**

© Zhigang Feng Redistribution prohibited without permission

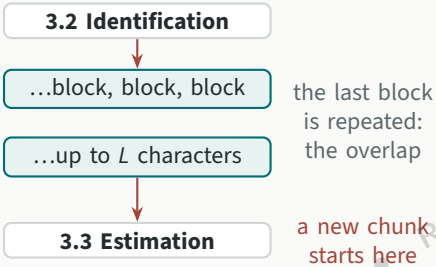
Stage 3, Chunking: Why Cut, and the Trap



- A **chunk** is the unit retrieval returns and the unit a citation names. Too small, it loses its context; too large, it mixes subjects and crowds the prompt.
- The trap is a claim cut from its condition: the sample, the specification, the date. Retrieved alone it reads as general, and is not.
- **Cut at the unit you would be willing to cite.**

© Zhigang Feng
Redistribution prohibited without permission

Stage 3, the Rule: Headings Are Hard Boundaries, and What One Chunk Carries



- Blocks under one heading accumulate to a target length L with a floor ℓ; the last block repeats into the next chunk as overlap, never a cut sentence, never across a heading.
- The working system: $L = 1200$ and $\ell = 200$ characters, overlap by whole blocks up to 150, read from its settings file.
- Each chunk carries its address: document, page range or section path, and its heading.
- **A chunk that spans two sections answers neither question.**

© Zhigang Feng
Redistribution prohibited without permission

Two Encoders, One Recipe: MiniLM for the Toy, BGE-M3 for the Library

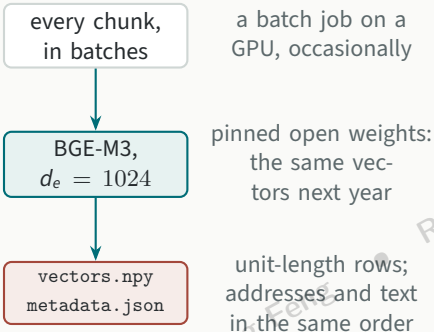
	the toy: all-MiniLM-L6-v2	**the library**: BGE-M3
built on	MiniLM: a 6-layer BERT, English	XLM-RoBERTa: 24 layers, 100+ languages
a passage becomes	384 numbers: its tokens averaged	1,024 numbers: its first token's vector
reads at most	256 tokens (11 of 1,264 passages cut)	8,192 tokens
trained on	over a billion pairs	pairs, after long-text pre-training
size; runs on	22.7 M parameters, 91 MB; a laptop	568 M parameters, 2.3 GB; GPU to build

- BGE-M3 has two more outputs, a weight per word (like a keyword score) and a vector per token. The build keeps only the one point.
- Change the toy's year: MiniLM keeps cosine 0.89, BGE-M3 0.80. Change its topic: 0.63 and 0.83. Neither replaces a filter on the address.
- **Vectors on disk are comparable only with the weights that wrote them: pin the file, not the name.**

Source: Model cards and configuration files of all-MiniLM-L6-v2 and BAAI/bge-m3, read 20 October 2026; Chen et al. (2024).

© Zhicheng Feng · Redistribution prohibited without permission

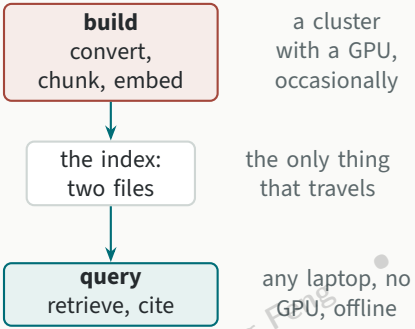
Stage 4, Embedding: One Open-Weight Multilingual Encoder, Vectors on Disk



- Choose the encoder on four things: languages, domain fit, passage length, and whether the weights are yours to pin. One model encodes corpus and question alike.
- A batch job on a GPU writes two files: unit-length vectors, and addresses and text in the same order. The build refuses vectors from any other model.
- **Open weights are a replication requirement: a hosted encoder can change under you; a pinned one re-runs.**

© Zhigang Feng. Redistribution prohibited without permission

Build on a GPU, Query Anywhere



- The expensive half runs rarely, where the GPU is. The half you use daily needs only the index and the encoder, on any laptop, with no network.
- Two environments on purpose: the query side never installs the GPU build of the deep-learning library. NumPy carries the store until the corpus passes a few hundred thousand chunks.
- **This split is the one design decision that makes a research RAG system livable.**

© Zhigang Feng Redistribution prohibited without permission

Stage 5, Query: Encode, Score, Rank, Hand Over

```
[doc-07 | title | p. 12 | 0.71]
§ 3.2 Identification
the passage, as converted, its heading
above it
[doc-12 | title | pp. 4--5 | 0.66]
…
[doc-03 | title | §2.1 | 0.64]
… k entries, best first
```

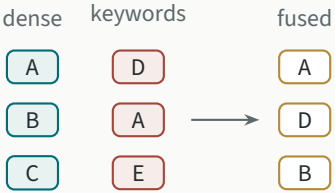
Answer only from the above. Cite each claim as [title, page]. If the evidence is weak, say so.

printed: the evidence, then the rule

- The question is encoded by the same model and scored against every vector in one product; the top k come back with their addresses. A topic filter applies *after* ranking, so a narrow literature never starves the result.
- The script prints evidence and an instruction. It never calls a language model itself.
- **Everything upstream exists to put good evidence in front of the model.**

Redistribution prohibited without permission
© Zhigang Feng

Two Upgrades When Dense Search Misses: Keywords, and a Second Reader



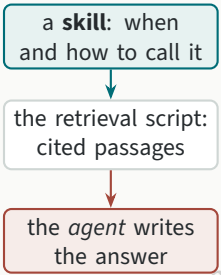
A tops one list and is second in the other, so it wins the merge

- Dense search can walk past an exact term; a keyword score (**BM25**) counts it. Merge the two lists by rank: $\mathrm{RRF}(d) = \sum_r 1/\big(\kappa + r(d)\big)$, $\kappa = 60$, a default fixed in a pilot. The lab's block B does this.
- A **re-ranker** reads question and chunk together, for the shortlist only; what the first pass missed it never sees.
- **Add these when a graded question set says the first pass misses, not before.**

© Zhigang Feng
Redistribution prohibited without permission

Source: Cormack, Clarke & Büttcher, *Proceedings of SIGIR '09* (2009), 758–759.

Make It a Tool the Agent Can Reach: A Skill



a second skill at ingest: read the front page, fill in author and year

- A skill folder tells the agent when and how to run the retrieval script; it reads it when a question needs the library.
- The agent writes the answer from the printed evidence: every claim tagged with title and page, or “insufficient evidence in the corpus”.
- **The generator is the agent you already have. The library is one more tool.**

© Zhigang Feng
Redistribution prohibited without permission

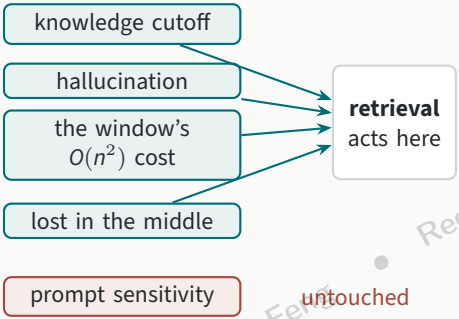
Where It Breaks, and Which Stage to Repair

What you see	Where it sits	What repairs it
nothing relevant comes back	the index	the corpus boundary
the right passage ranks fourth	the ranking	a second scorer, or re-ranking
a claim lost the condition it held under	the chunker	boundaries that respect structure
a confident answer about a year you never indexed	the query	a refusal rule
a confident answer from the wrong meeting	an ambiguous query	ask it back before answering

© Zhigang Feng Redistribution prohibited without permission

The lab produces three on purpose: a stale index, a cut inside the decision sentence, an ambiguous question answered from the wrong meeting. **The symptoms read alike from outside. Diagnose by asking which stage could have produced this one.**

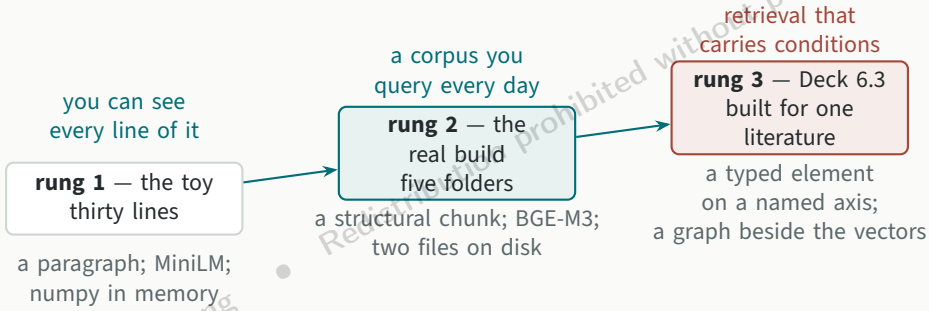
The Five Limits of Session 5, Four Repaired



- Session 5 closed on five limits of a pretrained model, all architectural: a larger model does not remove them.
- Retrieval repairs four by one move: the knowledge left the parameters for an index you can rebuild today. Prompt sensitivity is untouched; it belongs to the prompt, not the evidence.
- **Two remain: the refusal is a rule you write, and a dense vector cannot tell you when it is wrong.**

© Zhigang Feng Redistribution prohibited without permission

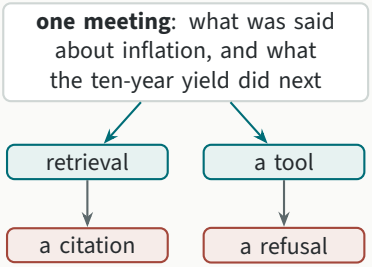
Three Rungs: Toy, Working, Research



© Zhengyuan Feng Reproduction prohibited without permission

- Each rung answers what the rung below answers badly, and costs more to keep standing.
- **Climb when the rung below is boring.**

Your Turn: Lab Blocks B, C and D



drop any one and the task is still answerable — fluently, and wrongly

- **B**, build the index on the 245 statements: chunks that name their statement, two lists fused by rank. **C**, answer or refuse by a fitted threshold, with three failures made on purpose. **D**, mark one answer's claims by hand.
- B opens by building this deck's toy, then hands you its five steps as one file for your own folder.
- **Self-paced, after class, on a CPU; the encoder is fetched once.**

Redistribution not allowed without permission
© Zhigang Feng

Source: The session's lab, blocks B to D; Session 2's frozen yield panel.