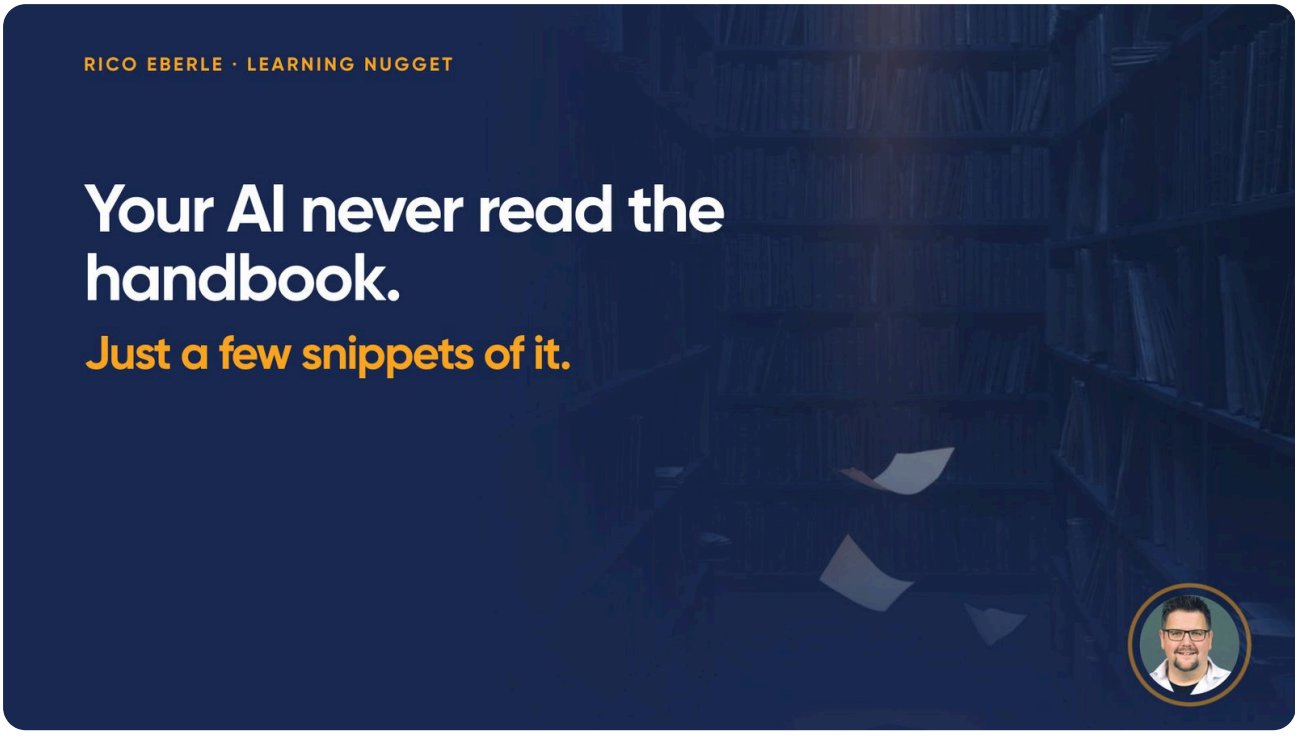
Rico Eberle · LEARNING NUGGETS

LEARNING NUGGET E-LEARNING & AI 1 OCTOBER 2026

RAG explained: your AI never read the handbook

An AI assistant with RAG only sees a few snippets of your documents; if the material fits in the context window, it is often better to put it all in.

By Rico Eberle, e-learning expert for continuing education and municipal councillor in Dübendorf



Video for this post

This learning nugget is also available as a video with transcript (1:44 min). Scan the QR code or open the link:

ricoeberle.ch/en/blog/rag-what-the-llm-sees/

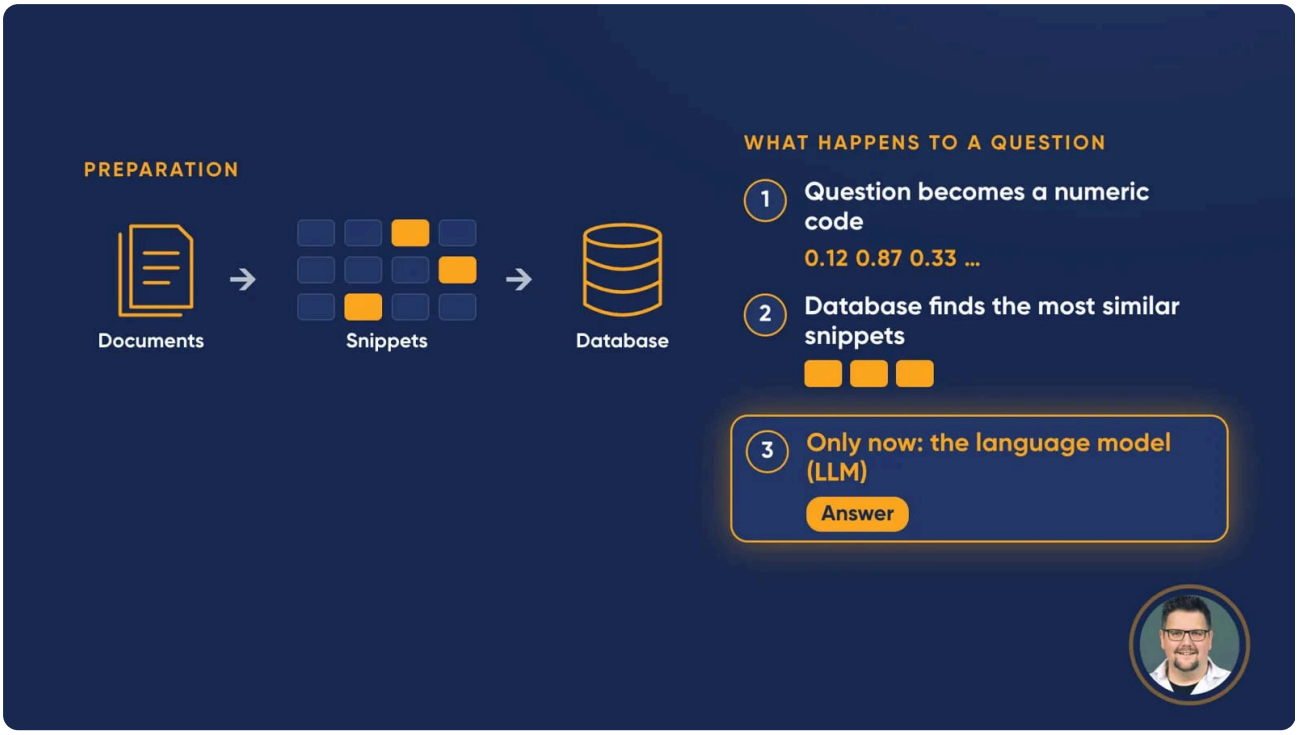
Many AI assistants for your own documents have never read the handbook. They only see a few snippets of it, picked by a search. The method is called RAG. Once you understand it, you can judge when such assistants answer reliably and when they don't.

How RAG works

RAG stands for retrieval augmented generation: text generation supported by a search. Patrick Lewis and colleagues described the method in 2020.

It starts with preparation. All documents are cut into small snippets. Each snippet gets a numeric code for its meaning and goes into a database. Every question then goes through three steps:

1. **Translate the question:** your question is also turned into such a numeric code.
2. **Search:** the database looks for the snippets with the most similar code, usually just a handful.
3. **Answer:** only now does the language model come in. It gets your question and these few snippets and writes the answer from them.



The video simplifies in a few places. In technical terms, the snippets are called chunks, the numeric code an embedding and the database a vector database. The embedding is also produced by a neural network, a so-called embedding model, but it isn't a language model that writes answers. Some systems also use a language model to rephrase the question or to re-rank the snippets found. How many snippets the model gets depends on the system.

The model only sees what the search finds

The language model never sees the whole handbook, only what the search picked beforehand. If the right snippet is missing, the model has nothing to stand on. Then it guesses, or says it can't find anything. Connections across chapters easily get lost, because each snippet stands on its own.

Anthropic described the problem in 2024: traditional RAG systems remove context when they split documents, which often means they fail to retrieve the relevant information. A sentence about revenue growth doesn't say on its own which company or which quarter it refers to. In Anthropic's tests, adding context to each snippet reduced such failures considerably, but didn't eliminate them.

The alternative: everything in the context window

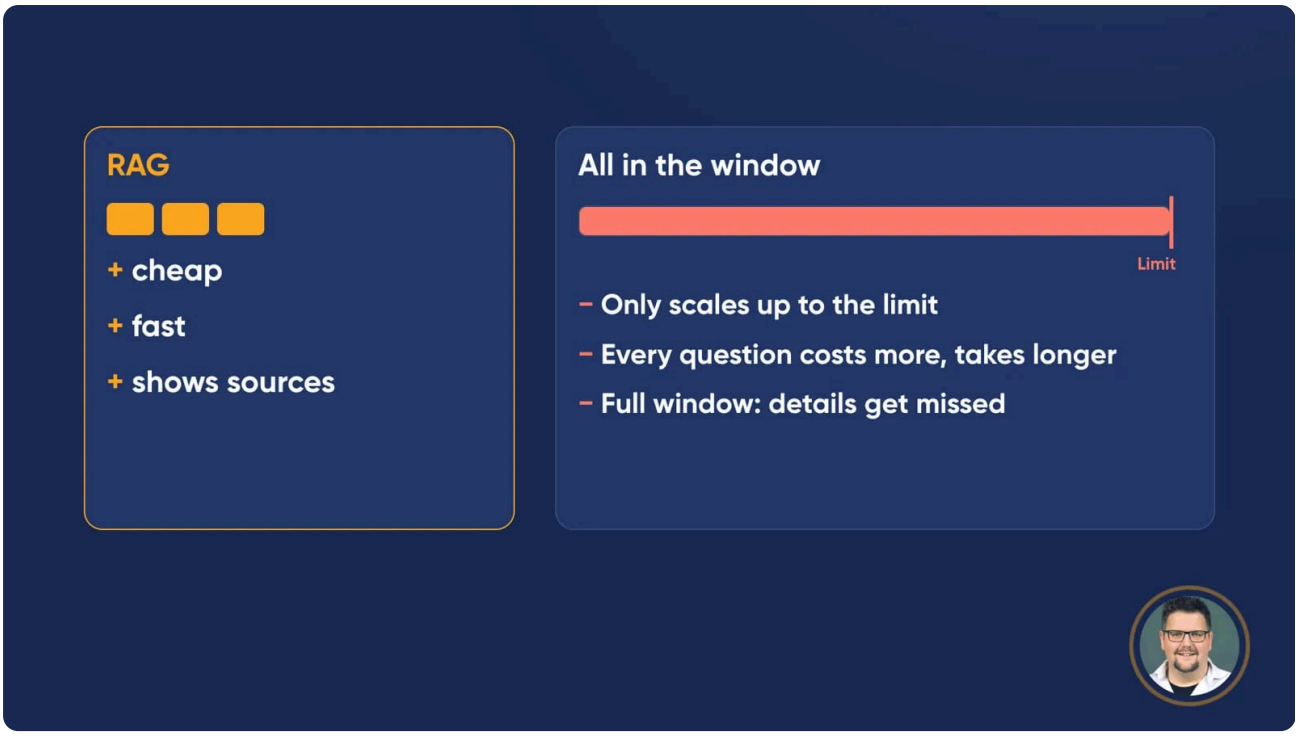
The context window is everything a model can take into account when answering, a kind of working memory. According to Anthropic, current Claude models hold a million tokens. Tokens are the pieces a language model breaks text into: words, parts of words or punctuation marks.

How many pages that is depends on the text. Anthropic equates 200,000 tokens with about 500 pages; scaled up, a million tokens comes to around 2,500 pages. That's a rough guide, not an exact conversion.

If you put everything in directly, the model really has it all in front of it. Studies show that this is often better. In 2024, Zhuowan Li and colleagues compared both approaches on several public datasets: given enough resources, long-context models performed better on average than RAG. Anthropic, too, recommends simply including the entire knowledge base in the prompt if it is smaller than 200,000 tokens.

The large window has limits

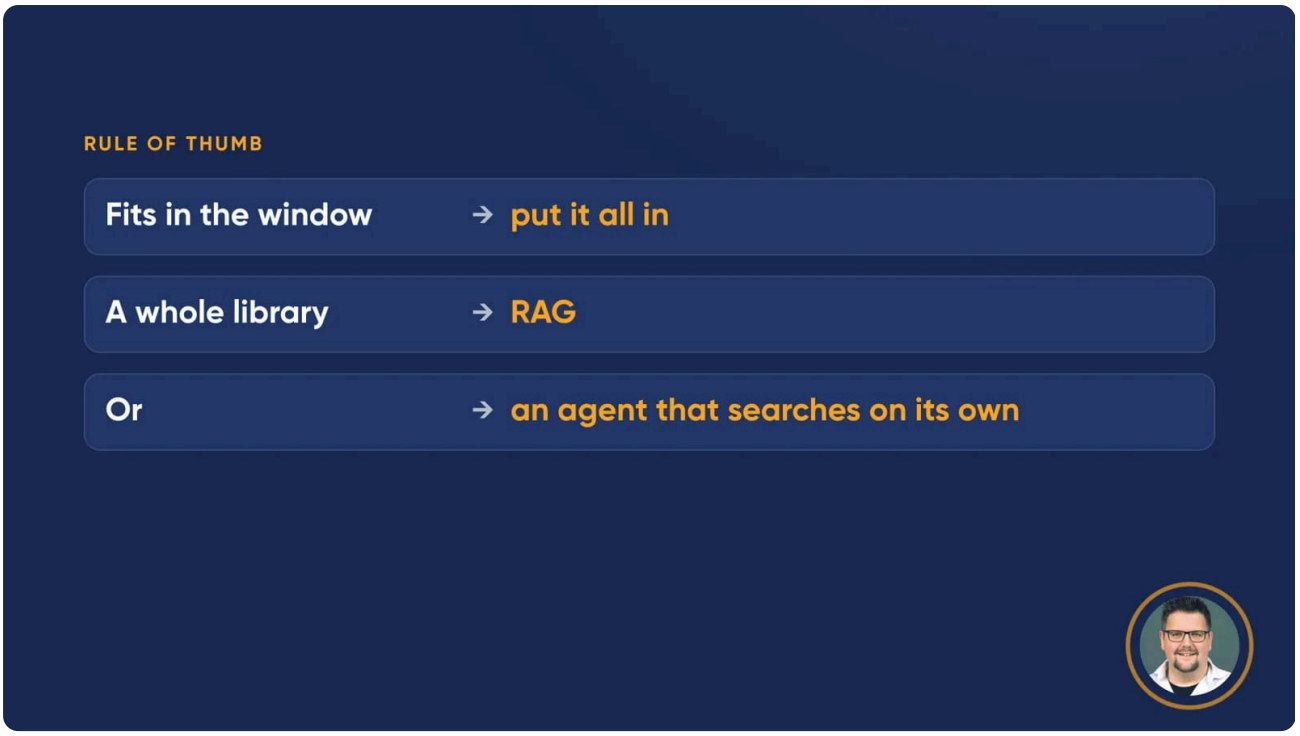
- **It only scales up to the limit of the window.** A whole library won't fit.
- **Every question costs more and takes longer.** Billing is per input token, and all the material goes in with every question. Prompt caching lowers the cost, but not to the level of RAG.
- **The fuller the window, the more likely the model misses details.** In 2023, Nelson Liu and colleagues showed that models make worse use of information in the middle of a long text than at the beginning or end ("lost in the middle"). In 2025, Chroma tested eighteen models and found that performance drops as input grows, even on simple tasks. Anthropic's own documentation calls this "context rot".



RAG, on the other hand, has real advantages. It is cheap and fast. And because it is known which snippets were used, it can show where an answer comes from. Lewis and colleagues already named the lack of provenance as a weakness of language models without retrieval.

The rule of thumb

If it fits in the window, put it all in. If it’s a whole library, you need RAG. Or an agent that searches on its own.



An agent doesn't pick snippets in advance. It searches with tools itself, checks what it finds and keeps searching if needed. Boris Cherny from the team behind Claude Code wrote in 2026 that early versions of the tool used RAG with a local vector database, but that agentic search generally works better and is simpler.

Check answers against the source

If you use AI assistants for your own material, such as course dossiers or regulations, a few things follow from this.

If an assistant shows which document an answer comes from, it's worth taking a look. That is one of RAG's strengths.

"Nothing found" doesn't mean "doesn't exist". The search may simply have missed the right snippet. Try asking again with different words.

Questions that span several chapters call for caution. Comparisons, summaries of entire documents or contradictions between sections easily get lost with RAG. If the document fits in the window, it is often better to provide all of it.

A short test with a document you know well shows how reliable a particular assistant is. Three questions are enough: a detail question answered in a single section, a question that links two chapters, and a question whose answer isn't in the document at all. Comparing the answers with the original quickly shows where the assistant is reliable. The results can also be worked through together with participants.

What exactly separates a model, a language model, a chat and an agent is explained in Model, LLM, chat, agent: four terms, one picture.

If you know what the model sees, you also know what it can’t know.

Sources

1. Lewis et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020)
   https://arxiv.org/abs/2005.11401
2. Anthropic: Contextual Retrieval (2024)
   https://www.anthropic.com/news/contextual-retrieval
3. Anthropic: Context windows, Claude documentation (2026)
   https://platform.claude.com/docs/en/build-with-claude/context-windows
4. Li et al.: Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach (2024)
   https://aclanthology.org/2024.emnlp-industry.66/
5. Liu et al.: Lost in the Middle: How Language Models Use Long Contexts (2023)
   https://arxiv.org/abs/2307.03172
6. Hong, Troynikov & Huber (Chroma): Context Rot (2025)
   https://www.trychroma.com/research/context-rot
7. Boris Cherny: Post on X about search in Claude Code (2026)
   https://x.com/bcherny/status/2017824286489383315
8. Anthropic: Building effective agents (2024)
   https://www.anthropic.com/engineering/building-effective-agents

ABOUT ME

Rico Eberle

I work in e-learning, hold a degree in business economics (FH) and sit on the municipal council of Dübendorf, Switzerland. I also chair the foundation board of WBK Dübendorf, a continuing education foundation. I make the learning nuggets on my own: topic, research, text and video. Questions and disagreement are welcome.

Contact: ricoeberle.ch · LinkedIn: linkedin.com/in/rico-eberle

LICENCE

Text, transcript and video by Rico Eberle are licensed under CC BY 4.0 (music and sound effects in the video excluded). You may share and adapt the content, including for teaching, as long as you credit the source. creativecommons.org/licenses/by/4.0/

SUGGESTED CITATION

“RAG explained: your AI never read the handbook”, Rico Eberle, ricoeberle.ch, 2026, https://ricoeberle.ch/en/blog/rag-what-the-llm-sees/, CC BY 4.0