Memory for LLMs
An IR Problem

Antonio Mallia

IR-RAG @ SIGIR'2025

July 17, 2025

About me

me@antoniomallia.it

www.antoniomallia.it

@antonio_mallia

amallia

in/antoniomallia

I am currently a **Staff Research Scientist** at Pinecone.

Prior to this, I served as an Applied Scientist on the Artificial General Intelligence (AGI) team at Amazon.

I hold a Ph.D. from New York University, where my research focused on efficient web retrieval methodologies.

Founded by Edo Liberty

Created the vector database category

Pinecone’s mission is to make AI knowledgeable

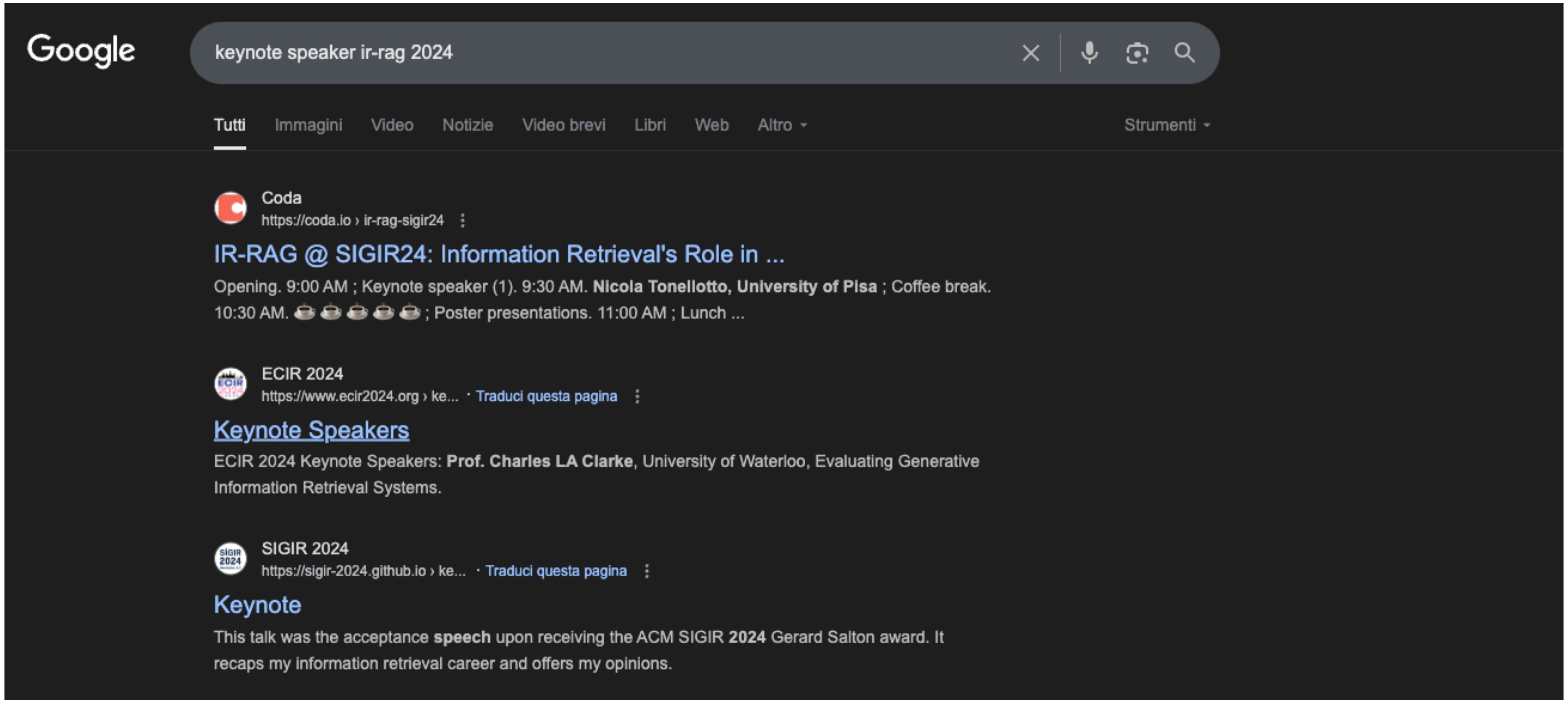
Pinecone is the leading vector database for building accurate and performant

AI applications at scale in production.

Go and check it out: https://www.pinecone.io/

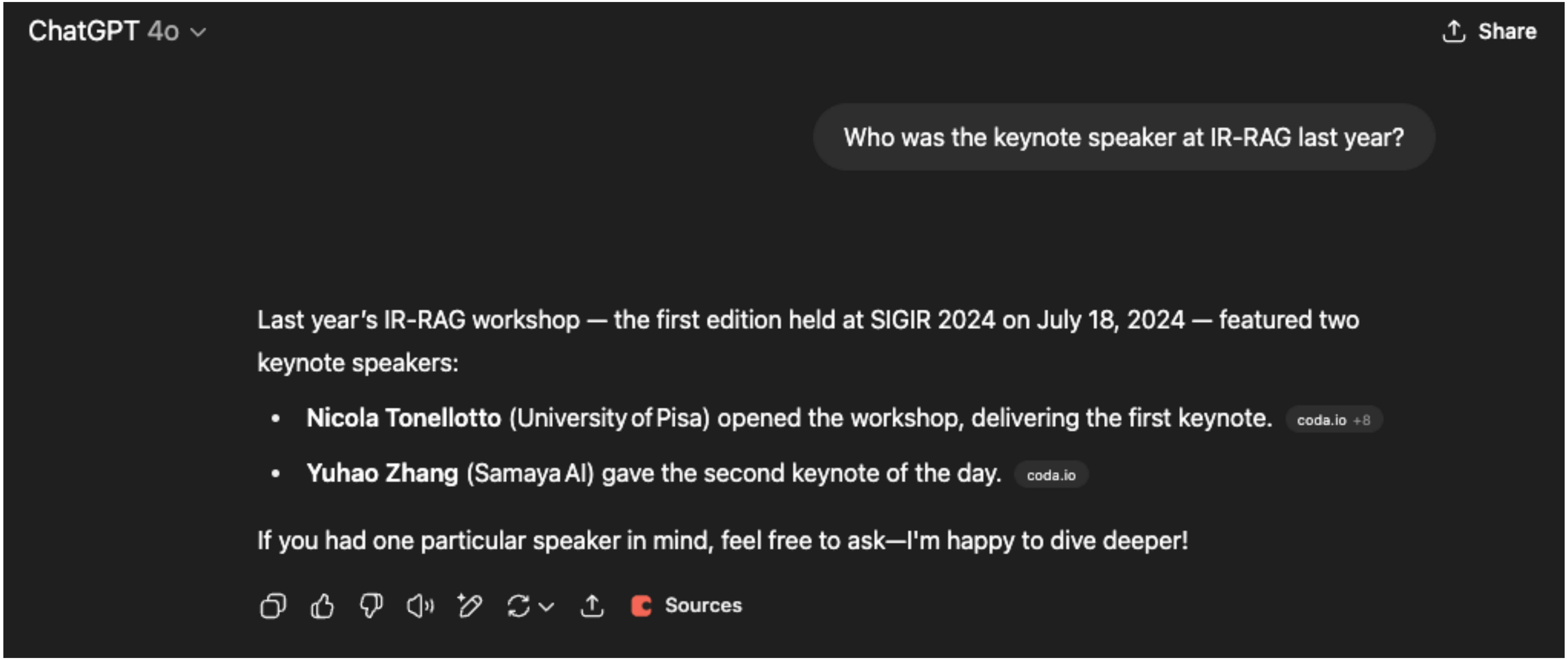
When we think about search

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections.



Nowadays, Information Retrieval goes beyond building just search engines!

When we think about RAG



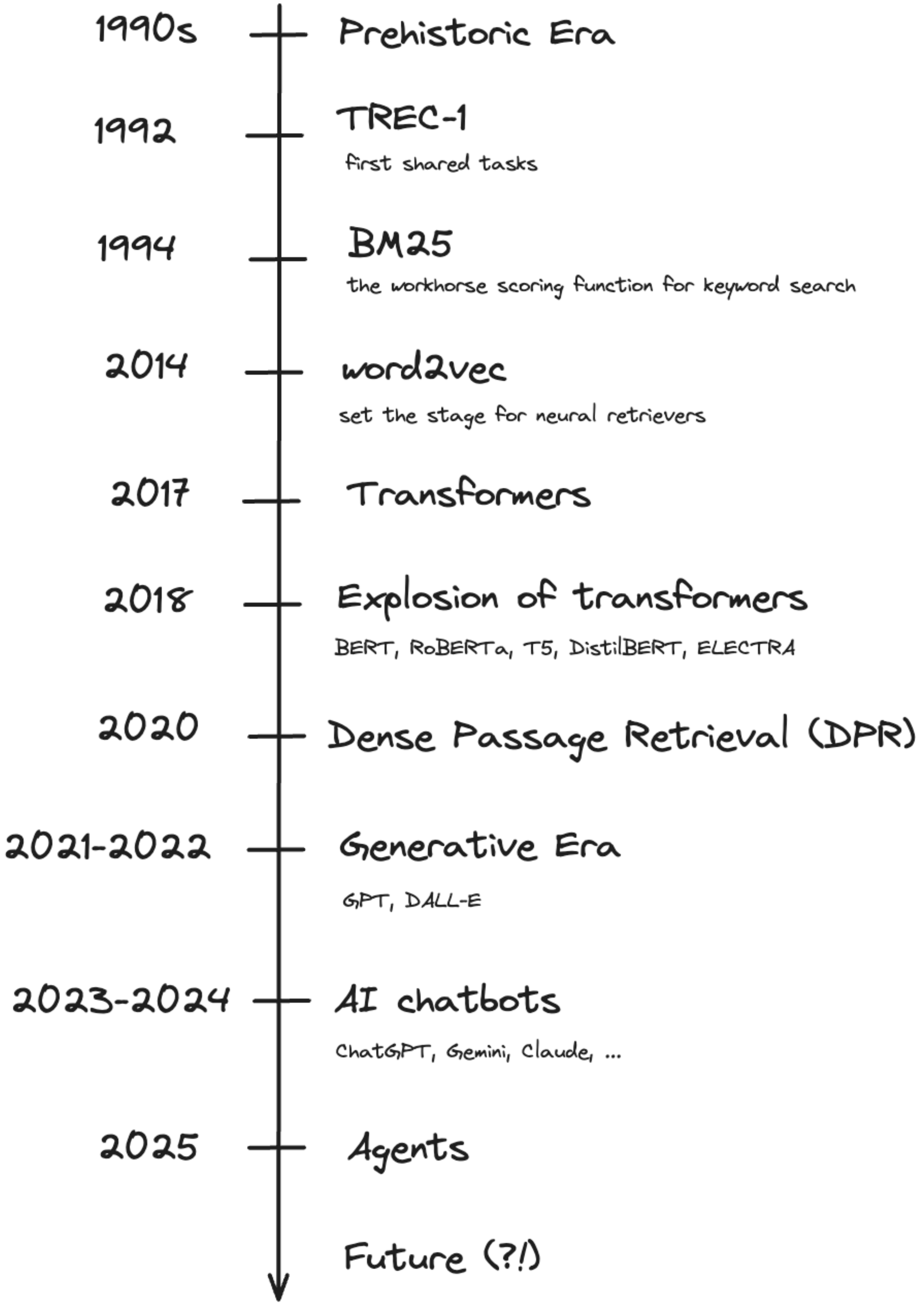
RAG Breaks the Old Rules

In RAG, the retriever isn't helping a user — it's helping a model. That changes everything.

Traditional IR	RAG Systems
User browses ranked list	LLM consumes retrieved content
Optimize for top-k relevance (MAP, NDCG)	Optimize for grounding and synthesis
Queries are short, keyword-based	Queries are natural, long-form, ambiguous
Focus on document ranking	Focus on passage evidence selection
Evaluation: clicks, rank positions	Evaluation: faithfulness, hallucination rate
Retrieval ends with the user	Retrieval fuels generation

From TREC-1 to Agents: 30 Years of Retrieval & Generation

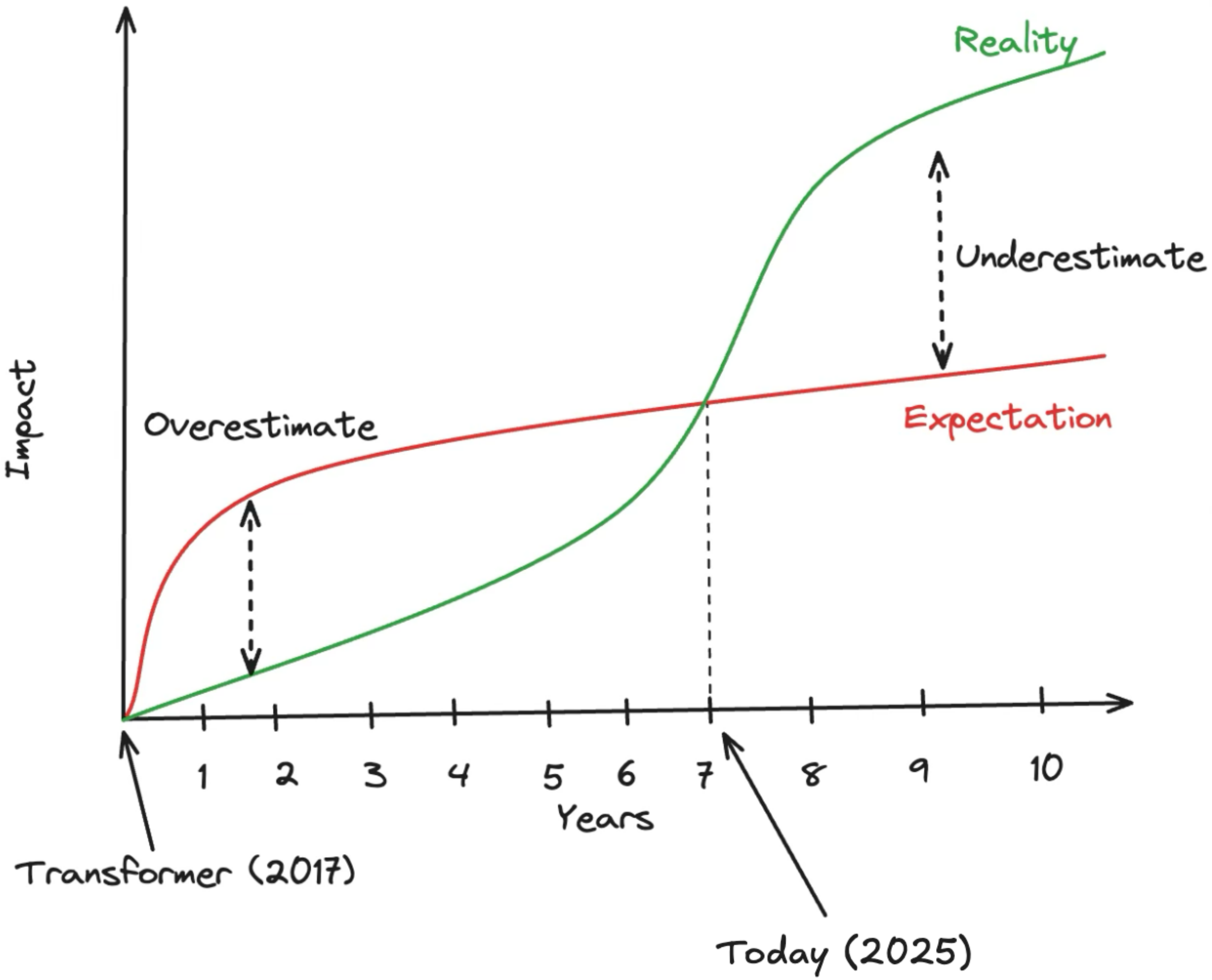
- Understand Progression
- Recognize Key Innovations
- Appreciate Impact on Applications
- Guide Future Research



Why Now?

We always overestimate the change that will occur in the next few years and underestimate the change that will occur in the next ten.

Bill Gates, The Road Ahead



Large Languange Models (LLMs)

Enhanced versions of the Transformers, often featuring millions or billions of parameters

Generally trained on vast quantities of diverse textual data, such as web corpora

They exhibit new capabilities as their scale increases, such as chain-of-thought reasoning

They require significant computational resources

Trained once, then adapted to various tasks without needing retraining

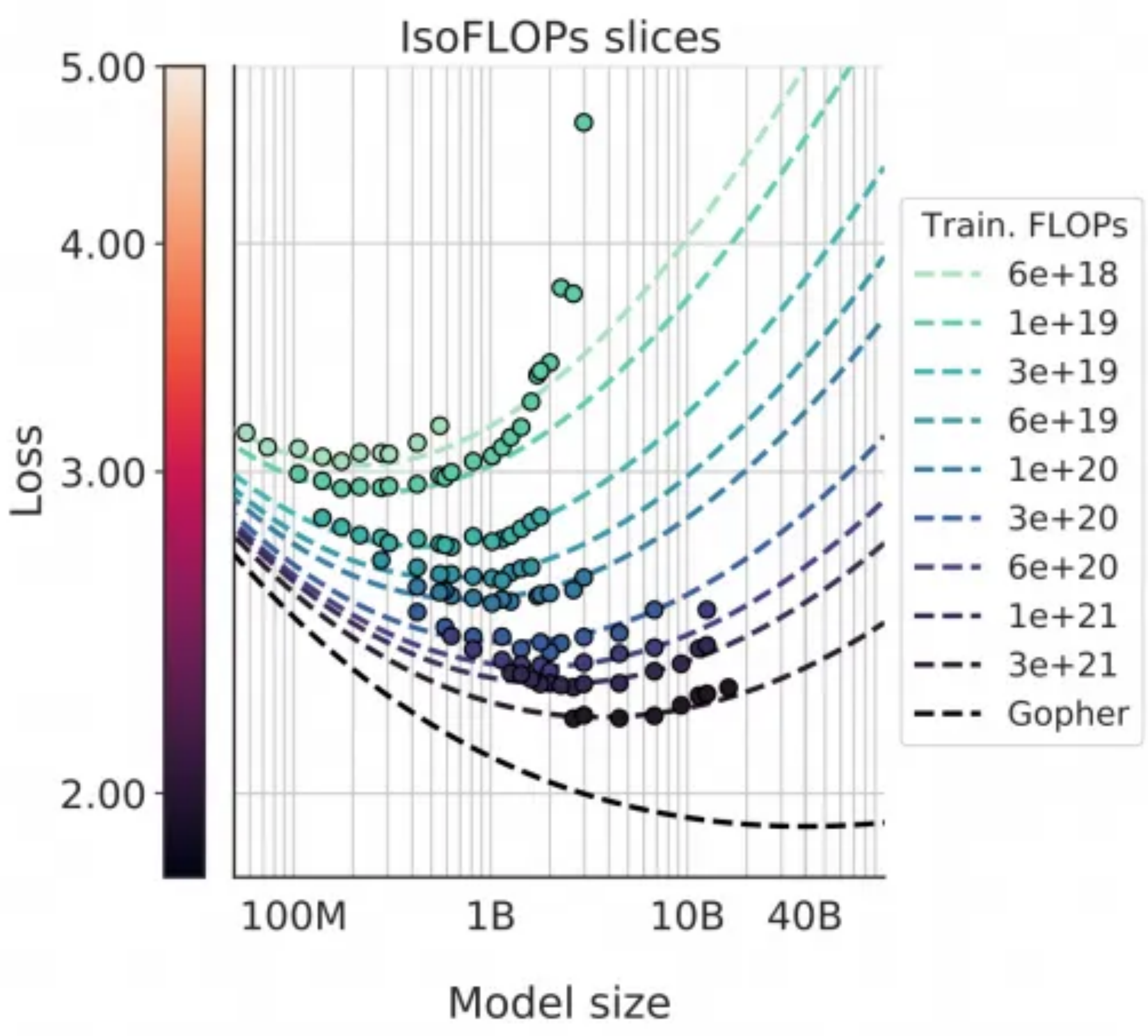
LLM Scaling Laws

The performance of an LLM is a function of:

•N — the number of parameters in the network

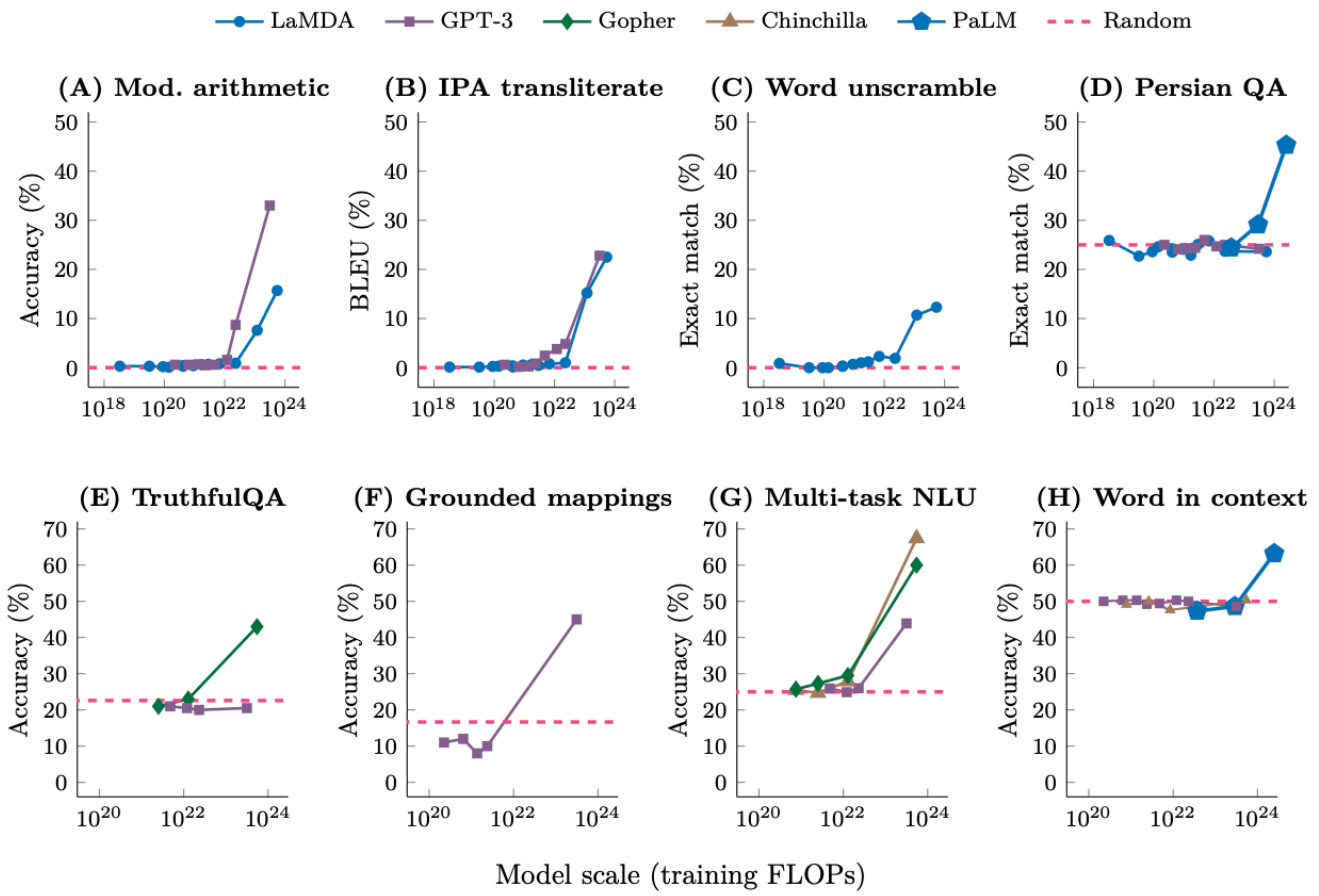
•D — the amount of text we train on

We get more "intelligence" for free with scaling!



Hoffmann, Jordan, et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022).

Emergent Abilities



Wei, Jason, et al. "Emergent Abilities of Large Language Models." Transactions on Machine Learning Research. 2022

Emergent Abilities

Emergent abilities may go beyond scaling!

Emergent abilities are also influenced by:

- new architectures
- higher-quality data
- access to external knowledge and tools
- enhanced training procedures

The future of LLM development may hinge on finding ways to enable smaller models develop new abilities


MEMORY

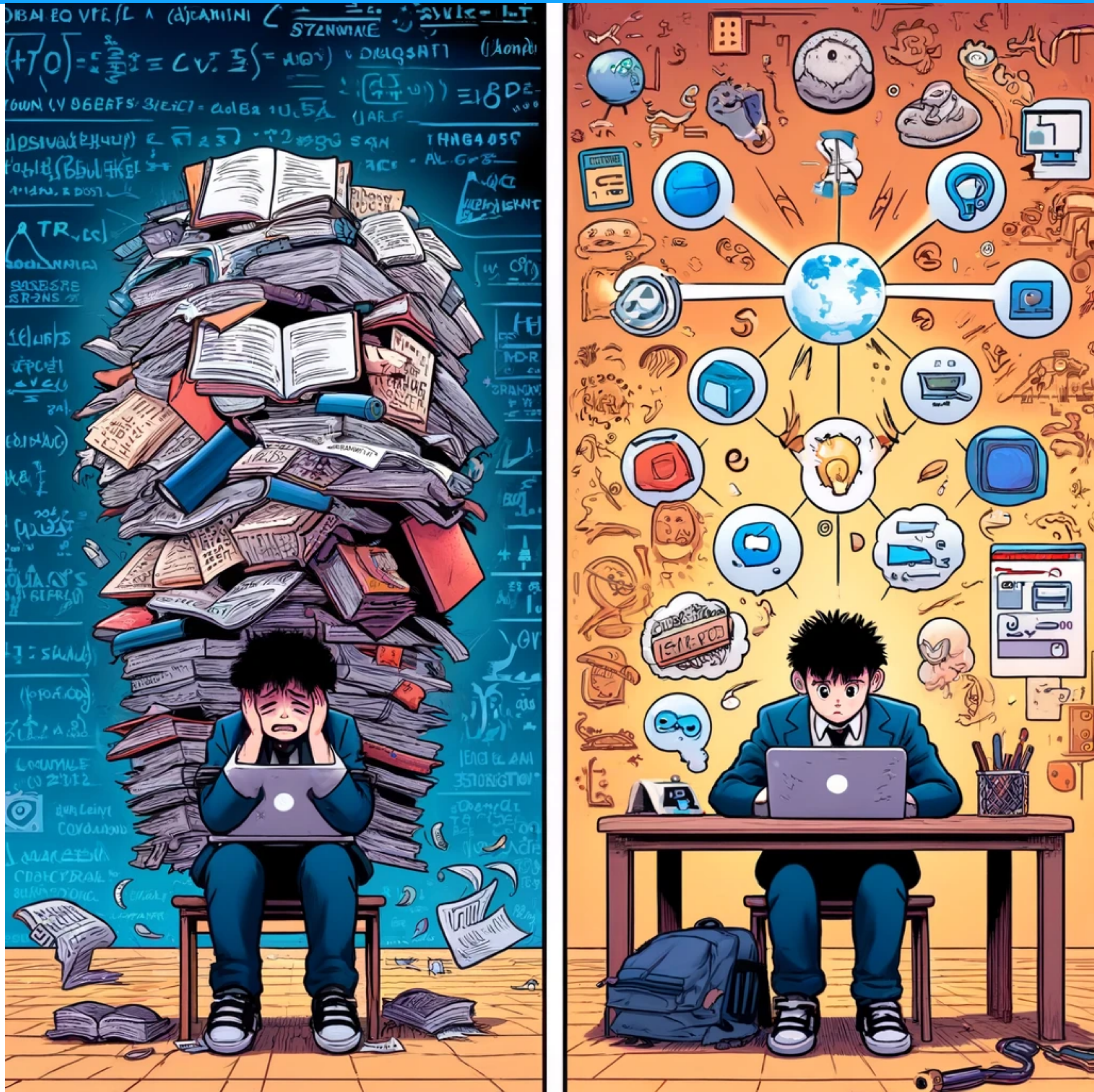
We Want Reasoning, Not Memorization

The model weights are serving dual purposes

Reasoning is the ability to pattern match across a diversity of input examples

The utility of **memorization** is actually relatively low

Choose a large model that can reason and memorize, versus just a small model that can reason - which do I pick?



Can long context replace the need for external memory?

"**Lost-in-the-Middle**" effect. Liu et al. (2023) show models lose ≥ 30 pp when the same answer-bearing sentence is moved from the start or end to the middle context.

The **BABILong benchmark** scatters one- or two-fact answers across 100k -10M token documents; popular long-context LLMs tap only 10–20 % of the window.

Retrieval still outperforms pure long context. RAG baselines in BABILong retain ≈ 60 % accuracy regardless of input length, highlighting that targeted retrieval + generation remains more reliable than simply extending windows.

input size \ Model	GPT-2 (137M)	mamba-2.8b-hf	rwkv-6-world-7b	v5-Eagle-7B-HF	Meta-Llama-3-8B-Instruct	LLaMA-2-7B-32K	longchat-7b-v1.5-32k	LongAlpaca-13B	Llama-2-7B-32K-Instruct	01-ai/Yi-34B	Mistral-7b-Instruct-v0.2	Mixtral-8x7B-Instruct-v0.1	01-ai/Yi-34B-200k	Mixtral-8x22B-Instruct-v0.1	activation-beacon-llama2-7b-chat	Yarn-Mistral-7b-128k	chatglm3-6b-128k	activation-beacon-mistral-7b	01-ai/Yi-9B-200k	Phi-3-mini-128k-instruct	ai21labs/Jamba-v0.1	c4ai-command-r-v01	Phi-3-medium-128k-instruct	GPT-4	~ Mamba (130M) fine-tune	Llama3-ChatQA-1.5-8B + RAG	~ RMT (137M) fine-tune	input size
0K	27	70	56	62	64	41	46	48	49	72	60	65	65	75	55	51	56	59	52	64	65	64	72	87	98	48	99	0K
1K	15	52	55	54	60	53	42	47	52	52	56	63	59	73	52	52	55	56	55	57	53	64	70	81	99	48	97	1K
2K		35	48	48	58	45	40	46	49	43	52	60	56	70	47	43	51	51	48	55	50	63	67	77	99	47	95	2K
4K		9	35	41	50	40	41	43	43	37	49	55	54	65	43	40	48	48	46	51	48	61	62	74	99	46	92	4K
8K		0	7	2	44	39	42	40	40	38	45	50	52	58	36	38	46	43	45	50	46	59	60	71	99	45	90	**8K**
16K						32	39	36	35	31	42	46	50	51	23	30	41	37	36	46	45	52	57	64	99	45	86	16K
32K						3	5	4	5	4	37	40	48	43	16	16	36	36	37	42	41	51	53	53	98	44	78	**32K**
64K													48	35	8	10	21	27	29	37	40	46	45	43	97	42	70	64K
128K															6	9	13	14	24	7	34	38	30	36	93	45	59	**128K**
512K																										42	46	512K
1M																										39	43	**1M**
10M																										37	34	10M

Kuratov, Yury, et al. "Babilong: Testing the limits of llms with long context reasoning-in-a-haystack." Advances in Neural Information Processing Systems 37 (2024).

Retrieval-Augmented Generation (RAG)

Combines retrieval of external information with generation capabilities of language models.

Incorporates up-to-date and specific information not contained within the model

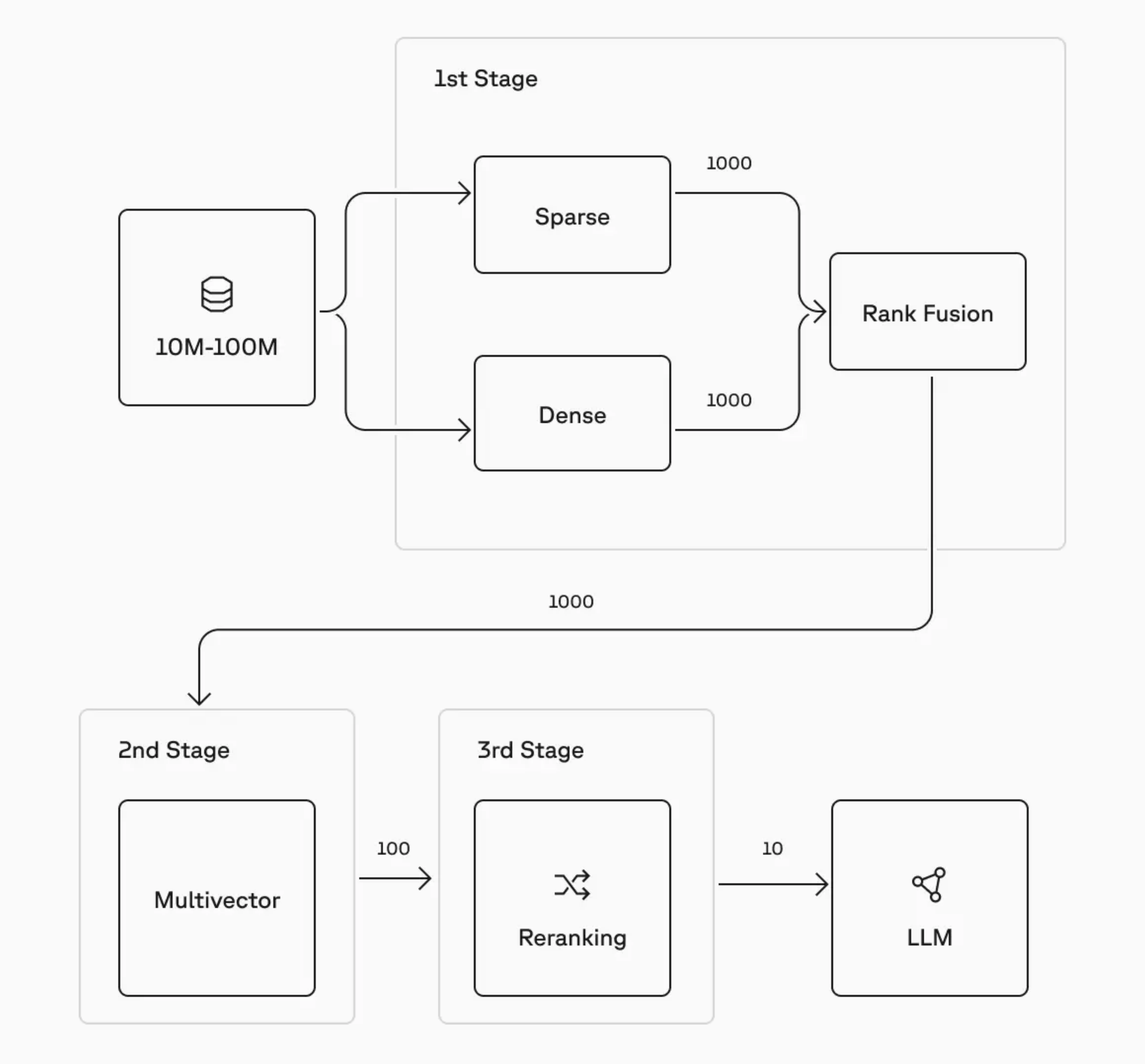
Reduces hallucinations by grounding responses in real data

Include citations to the original sources, enhancing credibility and traceability

Allows models to leverage vast external databases without the need for extensive retraining

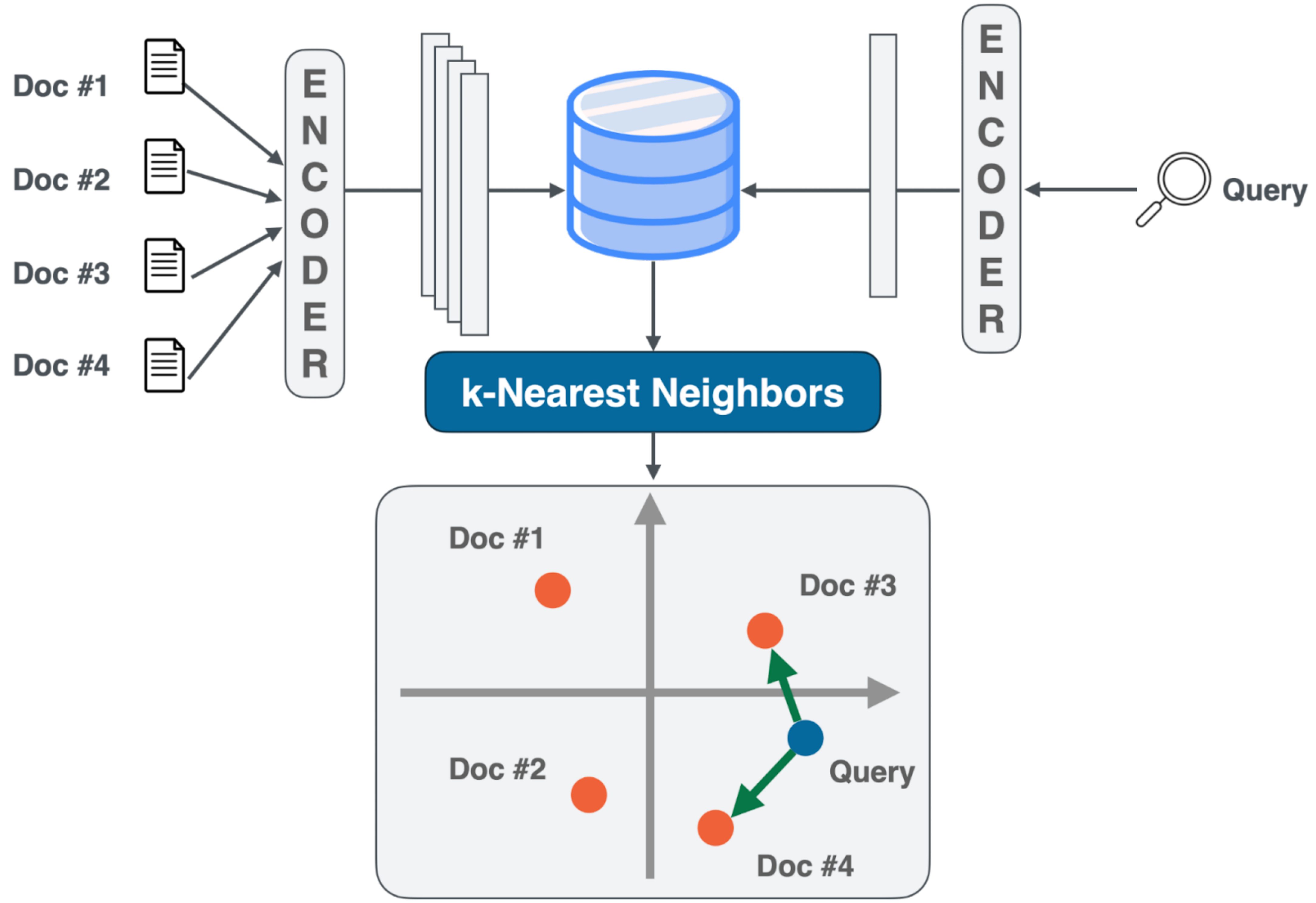
Model “does not know when it does not know”

A Standard RAG Pipeline



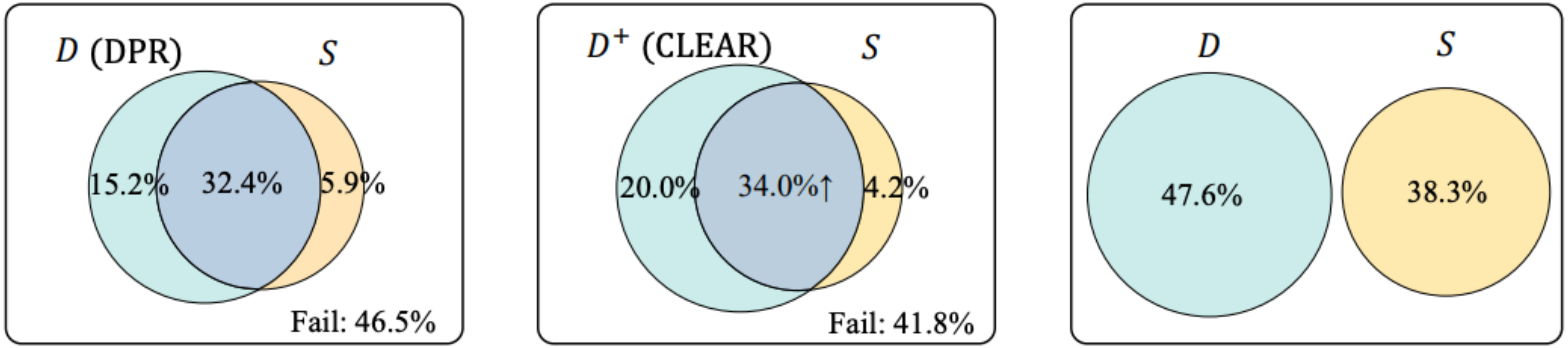
Sparse Retrieval

Semantic Similarity



Source: https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

Hybrid Retrieval

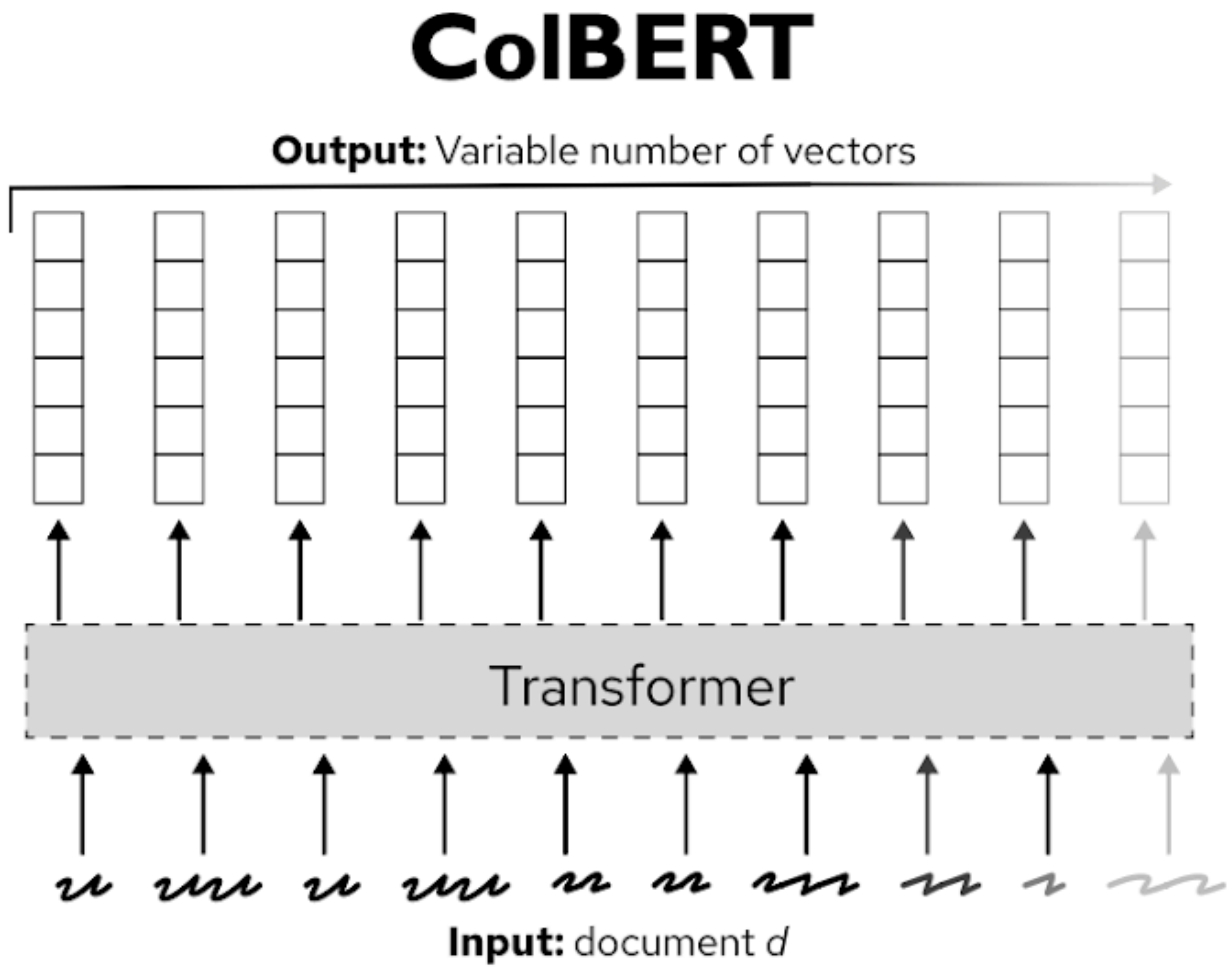


(a) D + S baseline (b) Hybrid Learning (CLEAR) (c) Hypothetical Scenario

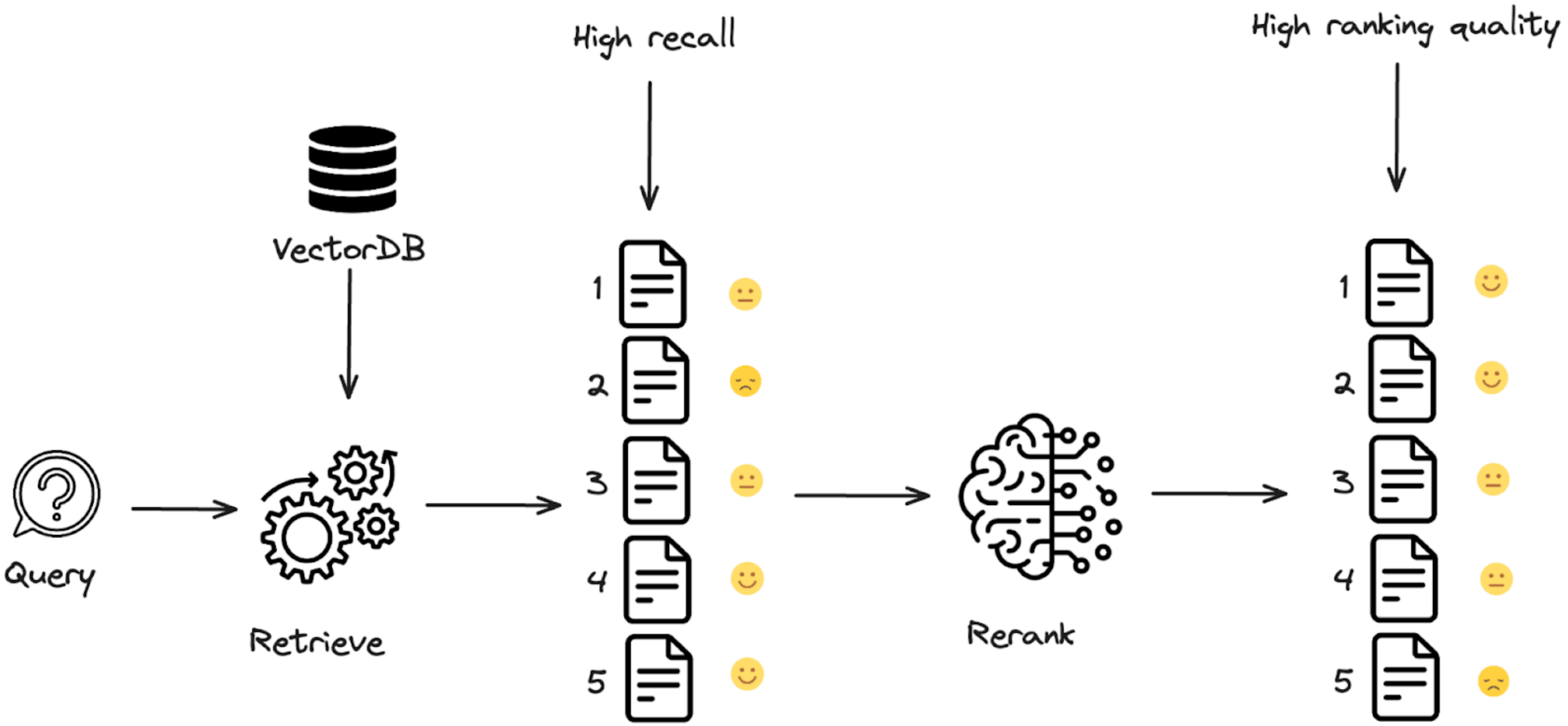
Figure 1: Recall@10 on Natural Questions. In the venn diargram, (a) shows BM25+DPR baseline and (b) shows CLEAR using residual margin. (c) is a hypothetical scenario, identical to (a) but without the intersection

On complementarity objectives for hybrid retrieval. Lee et al. ACL 2023.

Multi-vector Models



Reranking


High recall
High ranking quality
VectorDB
Query
Retrieve
Rerank
1
2
3
4
5
1
2
3
4
5

From Naive RAG to Advanced RAG

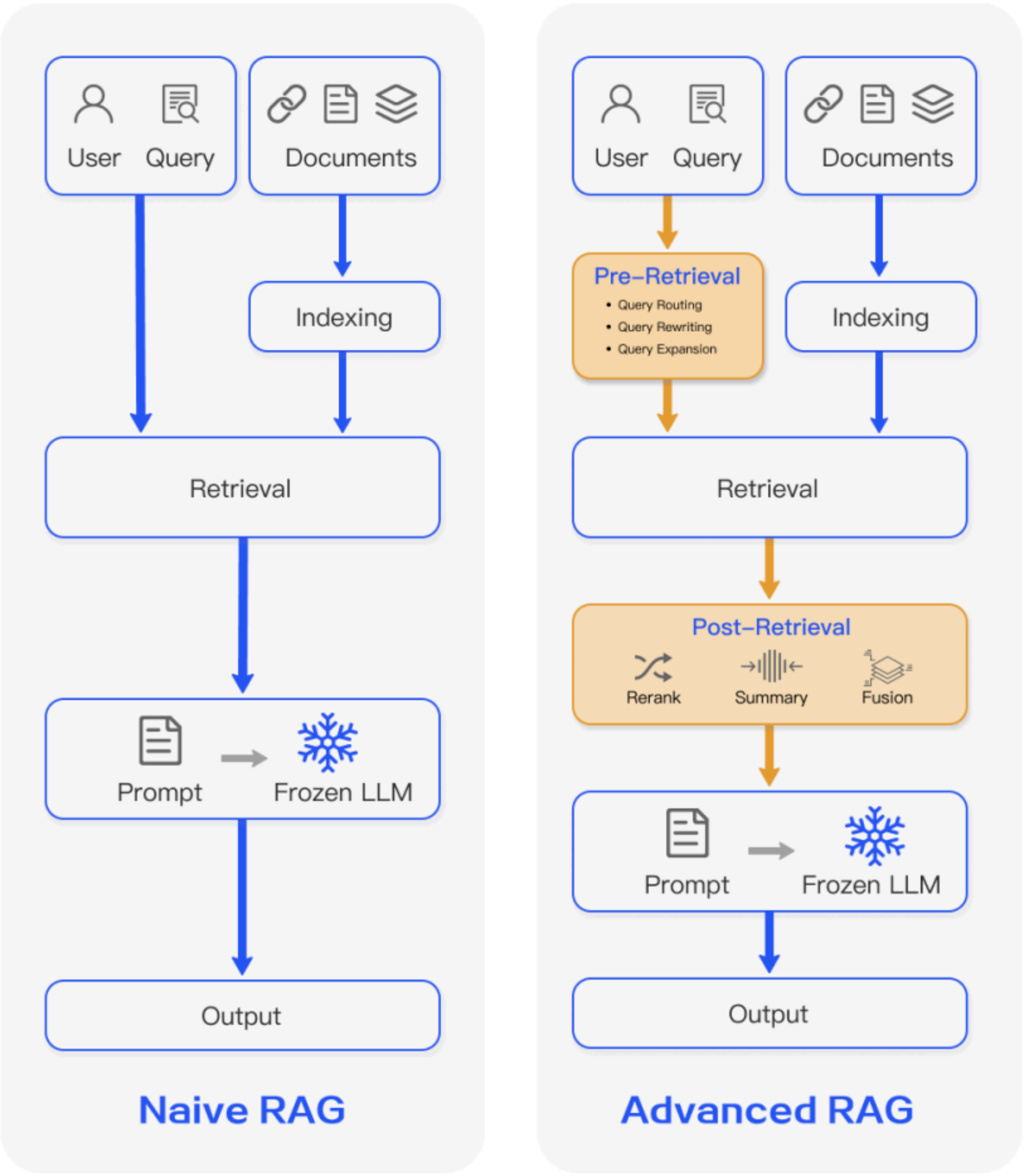
The gains are real

BUT

so are the trade-offs:

- latency
- engineering complexity
- infra spend
- operational risk.

At some point we need a new architectural leap rather than endlessly bolting on more patches.



Gao et al. "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv:2312.10997 (2023).

The Query-Length Explosion

Rewrite–Retrieve–Read–style rewriters expand a 5-word question into 40 – 80 token paraphrases.

Ma et al. "Query rewriting in retrieval-augmented large language models." EMNLP 2023.

HyDE fabricates up to 512 tokens per "pseudo-document" (often 8 per query) before the first retrieval call.

Gao et al. "Precise zero-shot dense retrieval without relevance labels." ACL. 2023.

Question-Decomposition (QD) RAG breaks one complex query into 3 – 6 sub-queries.

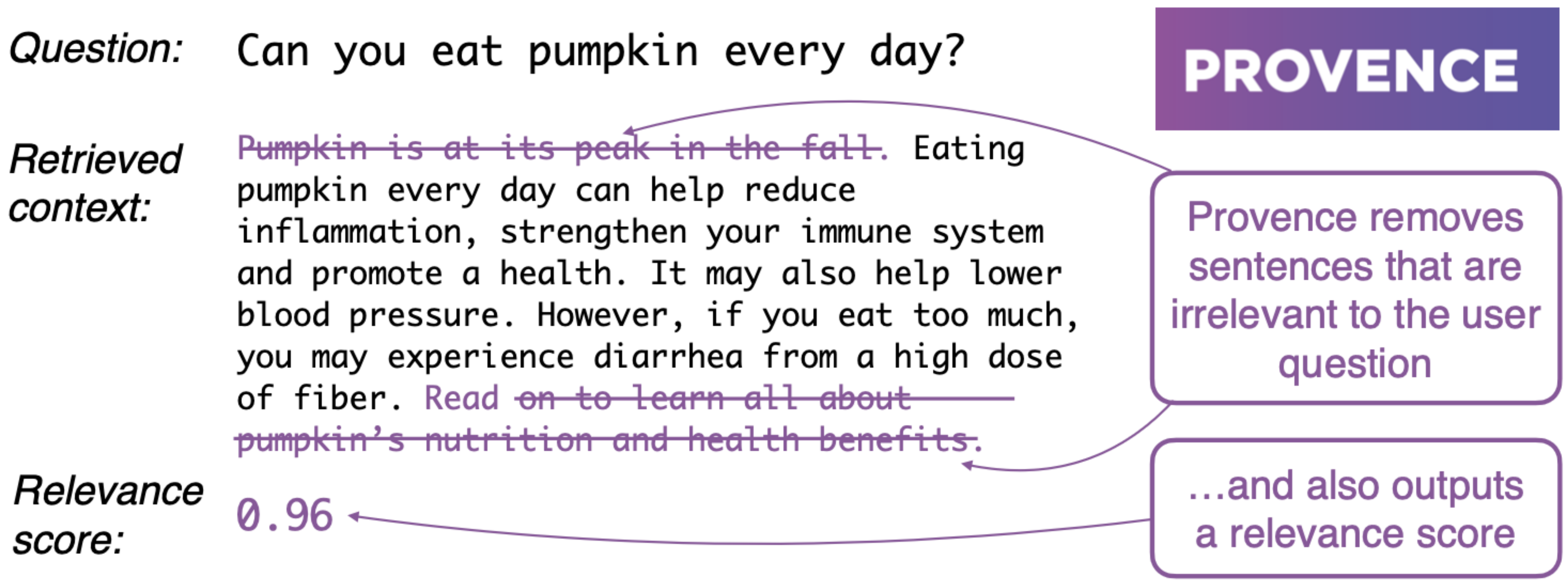
Ammann et al. "Question Decomposition for Retrieval-Augmented Generation." arXiv:2507.00355 (2025).

LevelRAG adds a high-level planner that iteratively rewrites until "coverage" is met.

Zhang et al. "LevelRAG: Enhancing Retrieval-Augmented Generation with Multi-hop Logic Planning over Rewriting Augmented Searchers." arXiv:2502.18139 (2025).

Context Pruning

Token costs shift left – the retriever now dominates the token bill



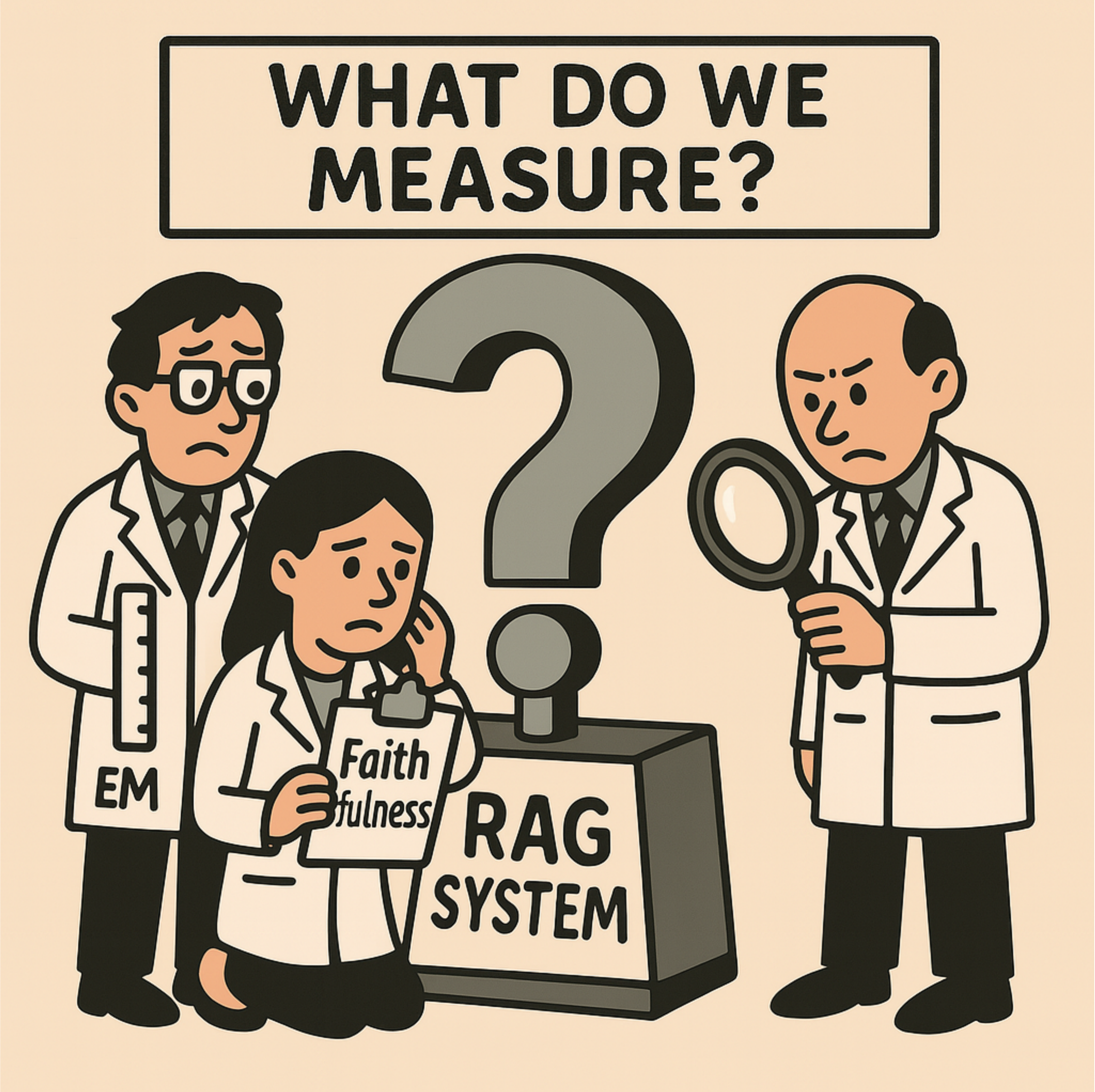
Chirkova, Nadezhda, et al. "Provence: efficient and robust context pruning for retrieval-augmented generation." arXiv preprint arXiv:2501.16214 (2025).

What Do We Measure?

Retrieval scientist holds a ruler labeled “**nDCG**”: classic IR relevance.

Answer scientist kneels with a clipboard “**EM** / **F1** / **BLEU**”: correctness.

Attribution scientist peers through a magnifier: **faithfulness** checks.



Hallucination by Distraction: Why Non-Relevant Docs Matter

Non-relevant ≠ neutral

Bad abandonment studies in classical IR show users quit when "**egregiously non-relevant**" results appear.

They distinguish plausibly vs egregiously off-topic docs, a distinction most RAG pipelines still ignore.

Moffat & Wicaksono, SIGIR '18

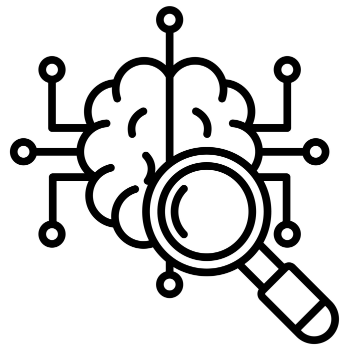
Accuracy can recover if the off-topic doc is replaced by pure random noise (dilutes distraction)

Cuconasu et al., SIGIR 2024

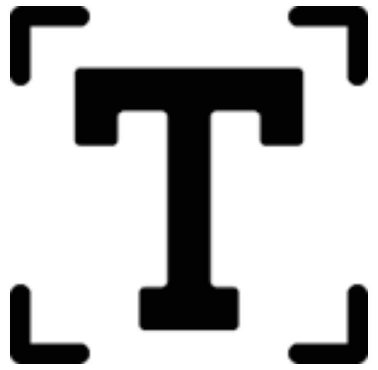
Reranker abstention: recent work adds a reject option for low-confidence passages instead of a rank order.

Gisserot-Boukhlef et al., arXiv:2402.12997 2024

Problems with RAG



Lack of seamless interaction between retriever and generator



Text as context is not *first-class citizen* for an LLM

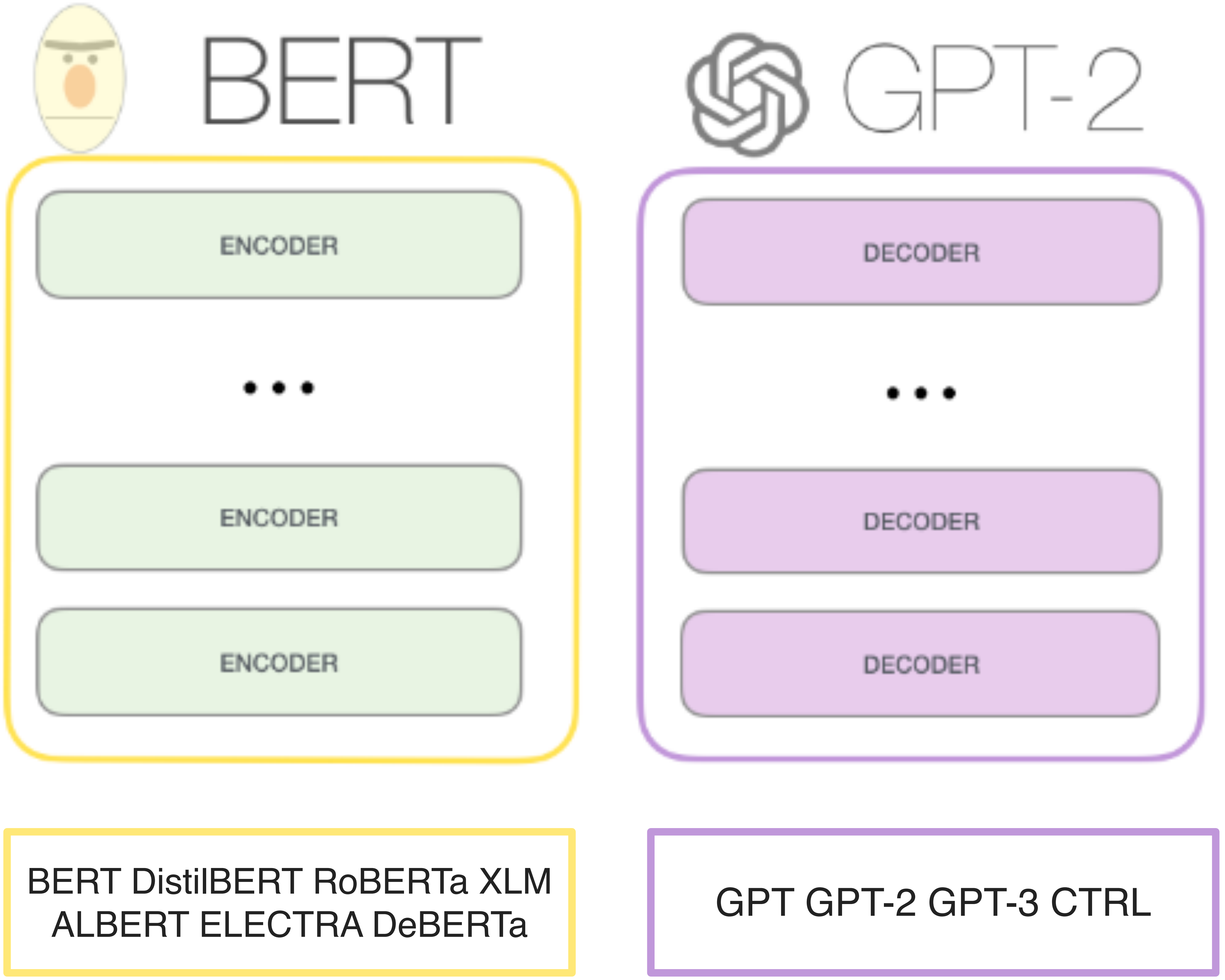
Corpus is almost static (minimal changes) processed by LLM over and over

Hard to define search relevance for an LLM as we do for users

Let’s go back to the LLMs

The Transformer

The original transformer model is composed of an **encoder** to process the input sequence and a **decoder** to generate the output sequence.



Encoder-only transformers can solve various NLU task

Trained using Masked Language Modeling (MLM) task.

Decoder-only transformers are mostly utilized for text-g

Trained using next token prediciton.

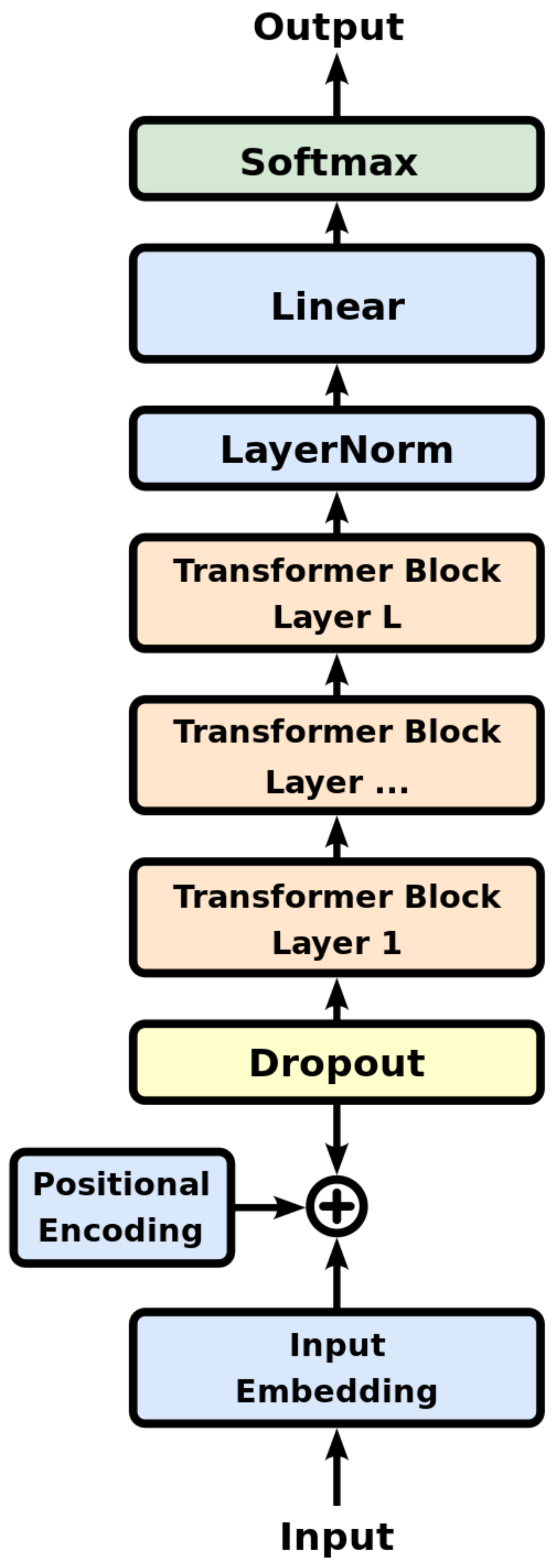
Generative Pre-trained Transformers (GPT)

GPT leverages the **decoder-only** transformer architecture

It consists of several **stacked transformer decoder layers**

GPT uses positional encodings to inject information about the relative position of tokens in the sequence

A softmax is applied to projected output to obtain a **probability distribution over the vocabulary**



Decoder Transformer Block

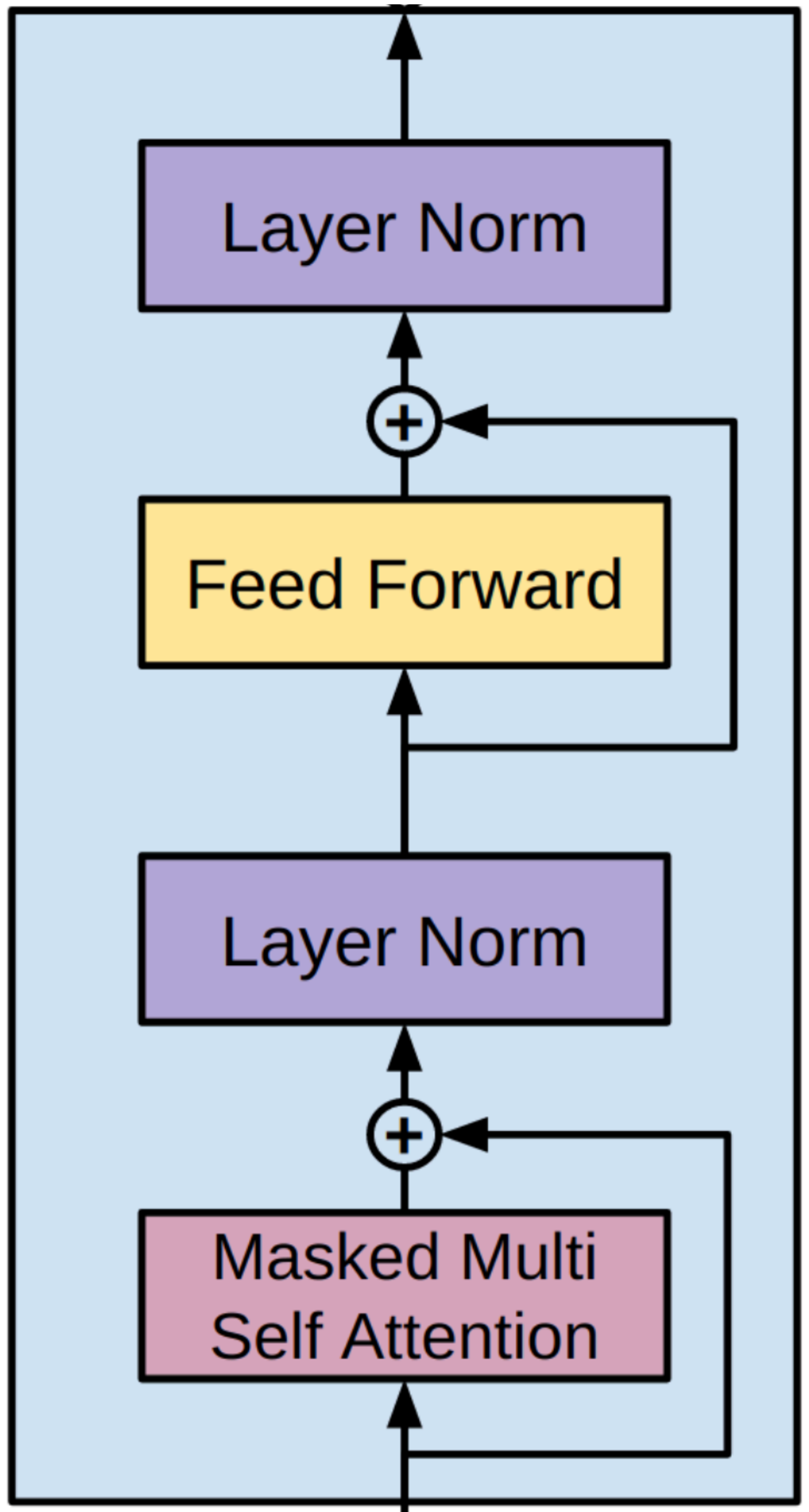
A GPT model is composed of multiple Decoder Blocks stacked on top of each other, forming **layers**.

The depth of the model, determined by the number of these layers, directly influences its capacity

The **masked self-attention** allows each position in the input sequence to attend to all other positions and ensures that the prediction for a given token only depends on known outputs.

The **feed-forward neural network** helps in learning complex patterns and representations

Each sub-layer is followed by **layer normalization** and residual connections, who help stabilize training and improve gradient flow



Improving language understanding with unsupervised learning - https://openai.com/index/language-unsupervised/

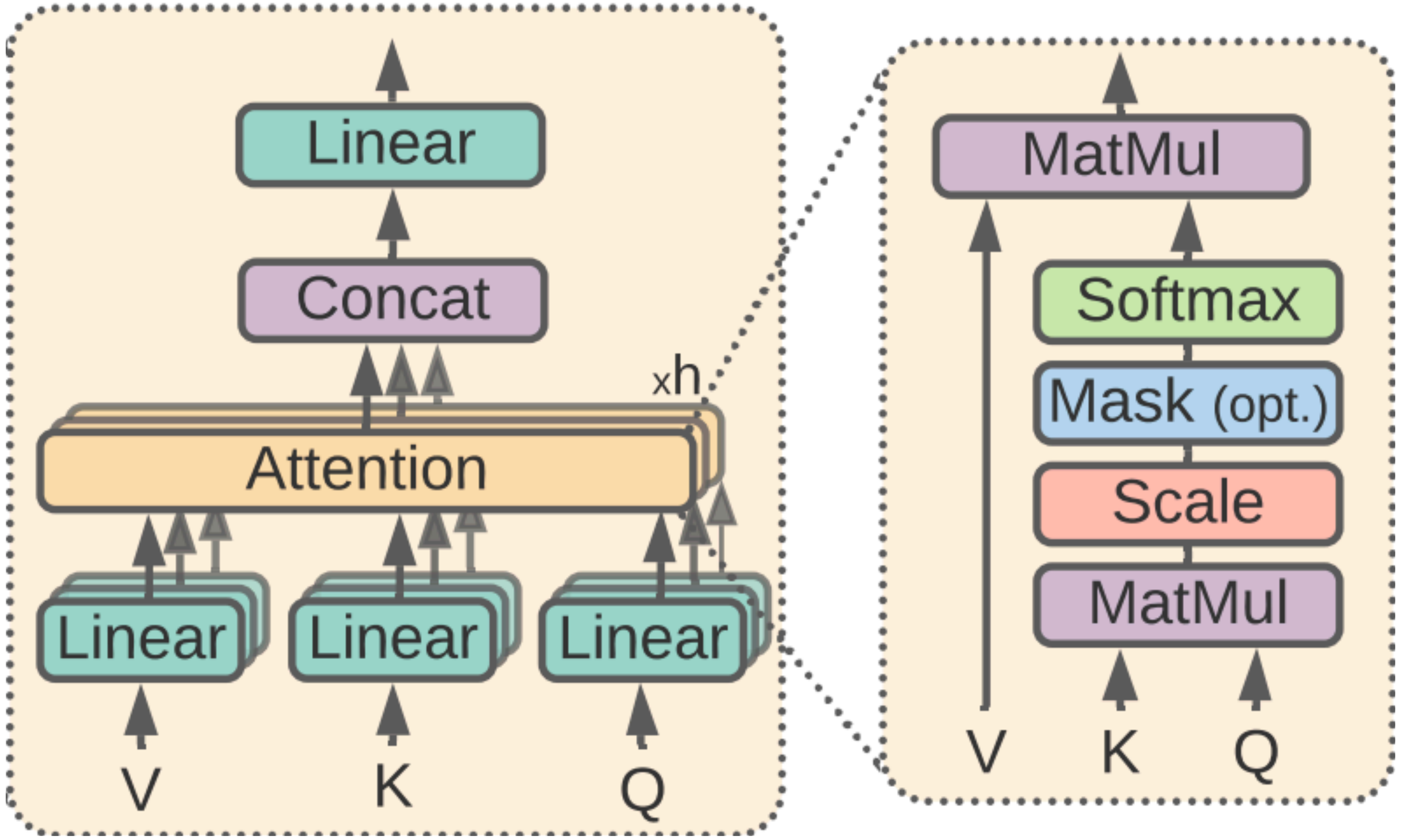
Attention Mechanism

It captures dependencies and **relationships between words** in a sentence

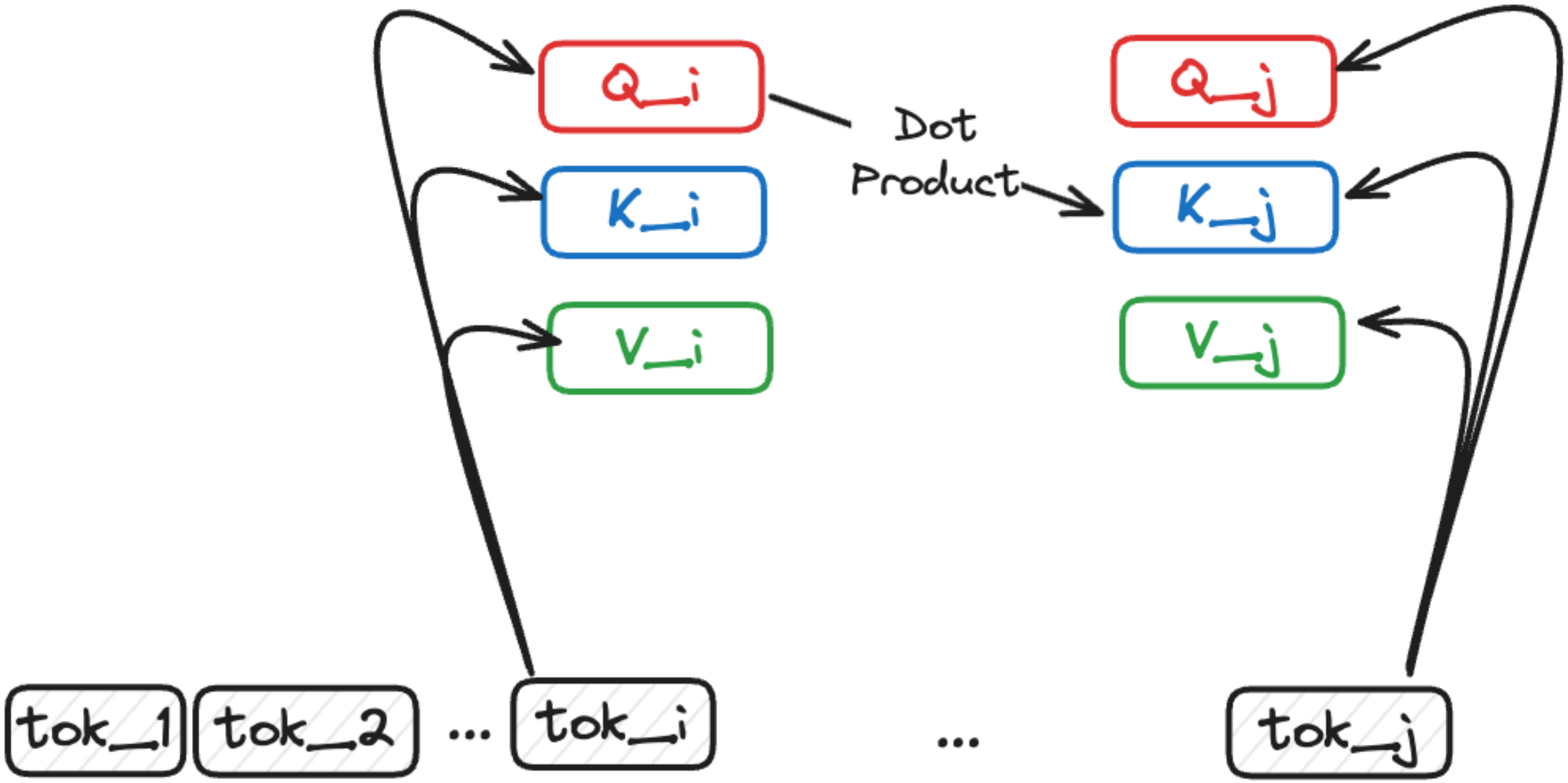
Multi-Head Attention is an extension of the attention mechanism where multiple "heads" operate in parallel

The attention mechanism involves computing **attention scores** using queries (Q), keys (K) and values (V)

The outputs from several attention heads are concatenated and processed through a linear layer



Attention Computation

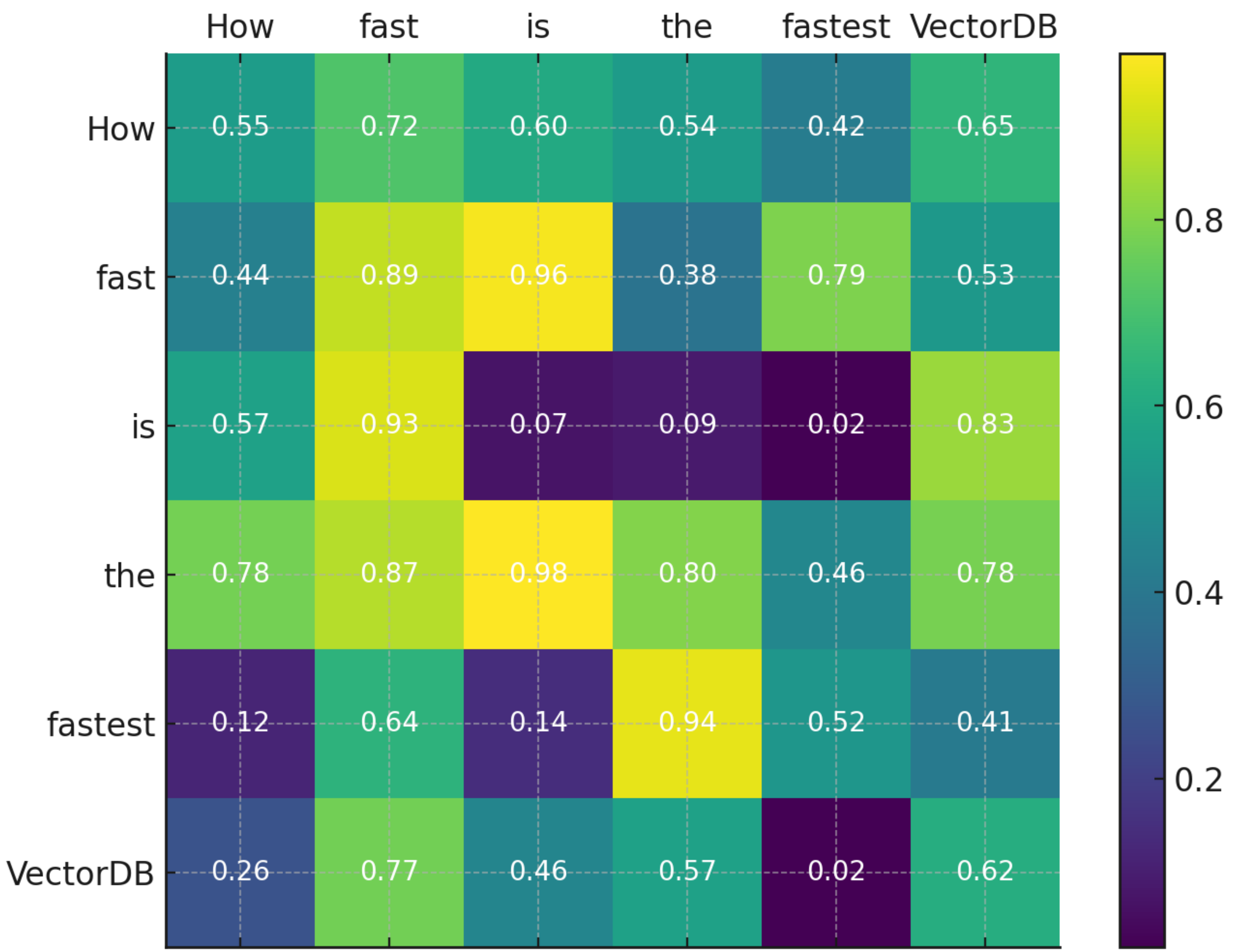


Attention Matrix

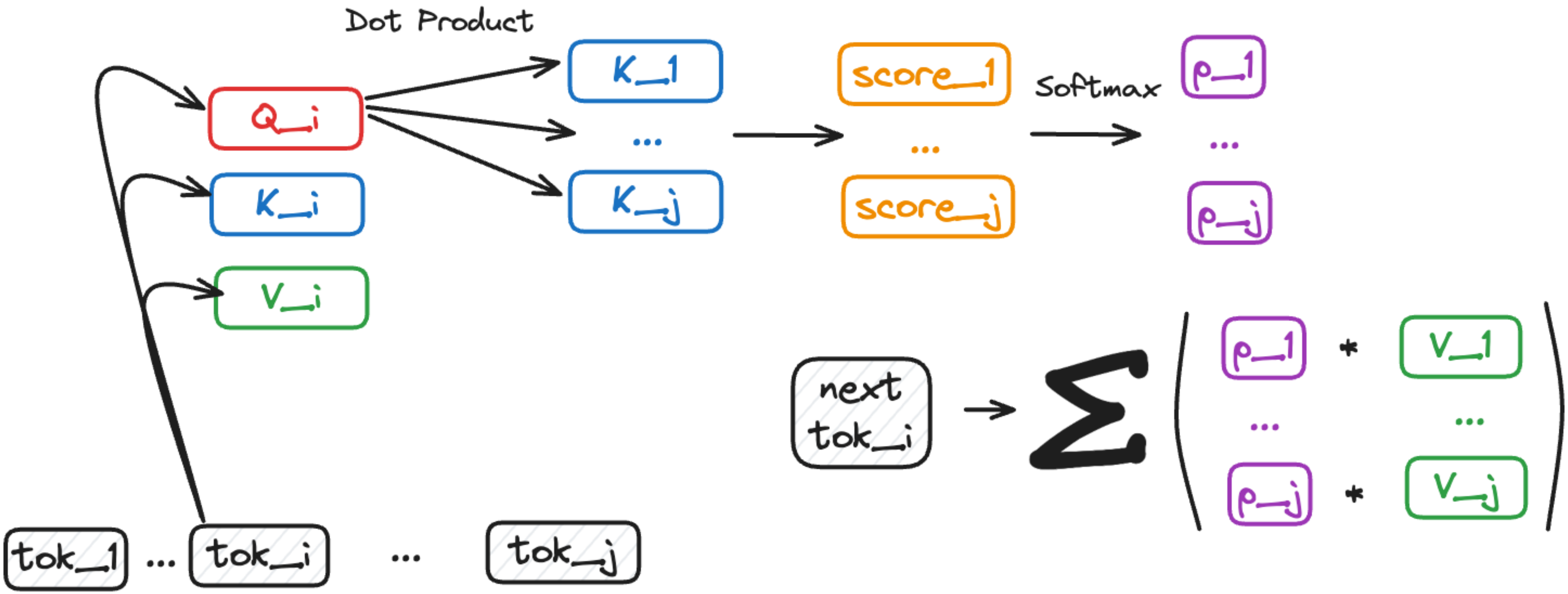
How much **focus or attention** each word gives to every other word when processing the sentence

Each value represents the **attention score** between the corresponding row and column words

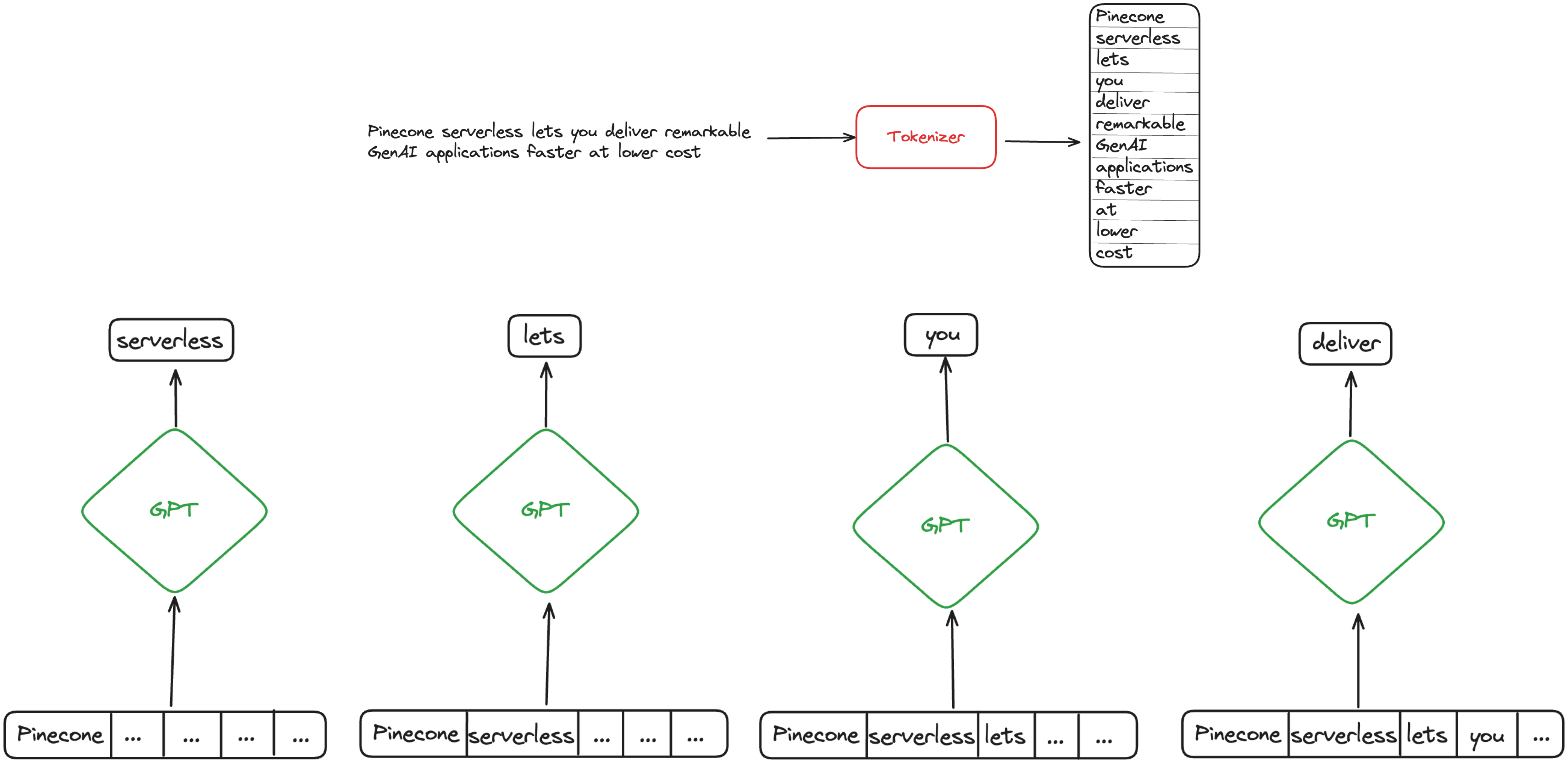
"fast" pays high attention to “is”, conversely "is" pays minimal attention to “fast” showcasing the **directional** nature of attention



Next Token Score



Next Token Prediction



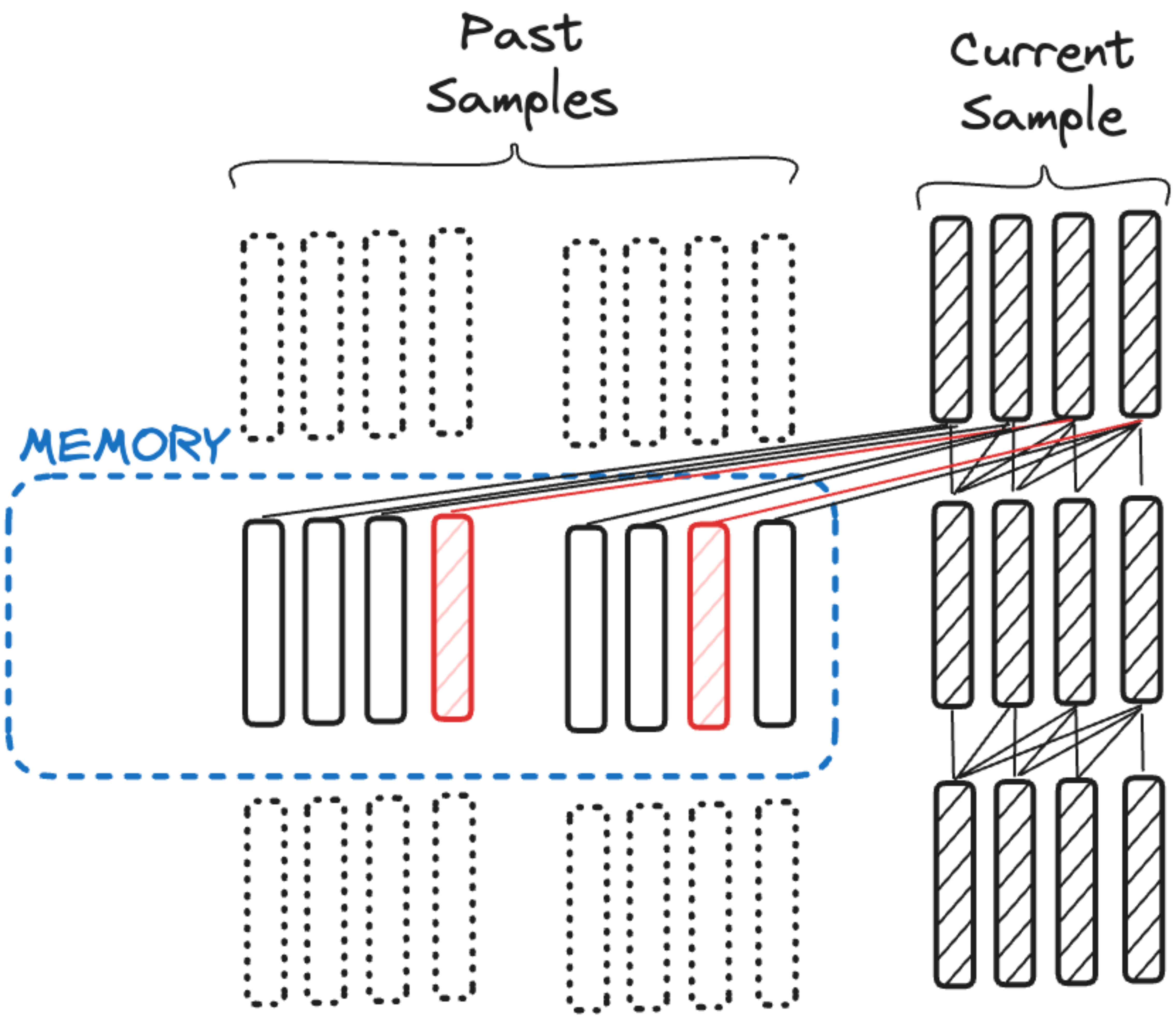
Memorizing Transformers

External Memory: Extends a standard Transformer with external memory.

kNN Lookup: During inference, the model performs an approximate k-nearest-neighbor search in the memory to retrieve relevant past contexts by matching keys.

Leverage past knowledge: attends to both the normal sequence context and the retrieved memory key–value pairs

Extended Context: Effectively gives the model a much longer memory beyond the fixed context window.



Wu, Yuhuai, et al. "Memorizing Transformers." International Conference on Learning Representations.

Other Attention Mechanisms

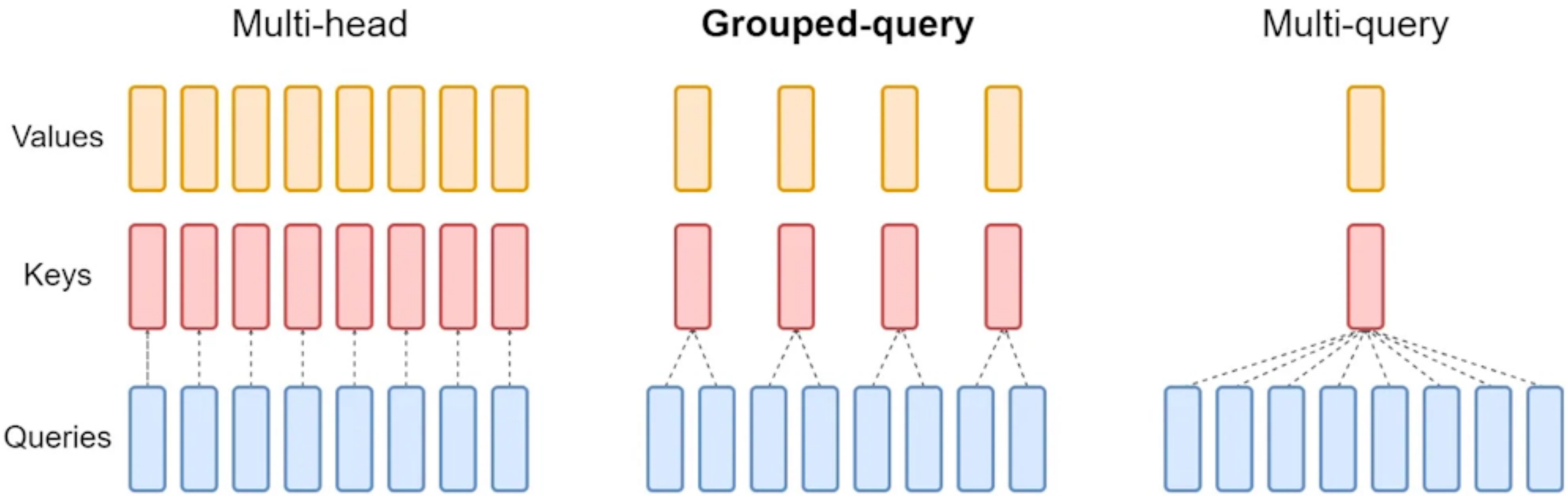


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

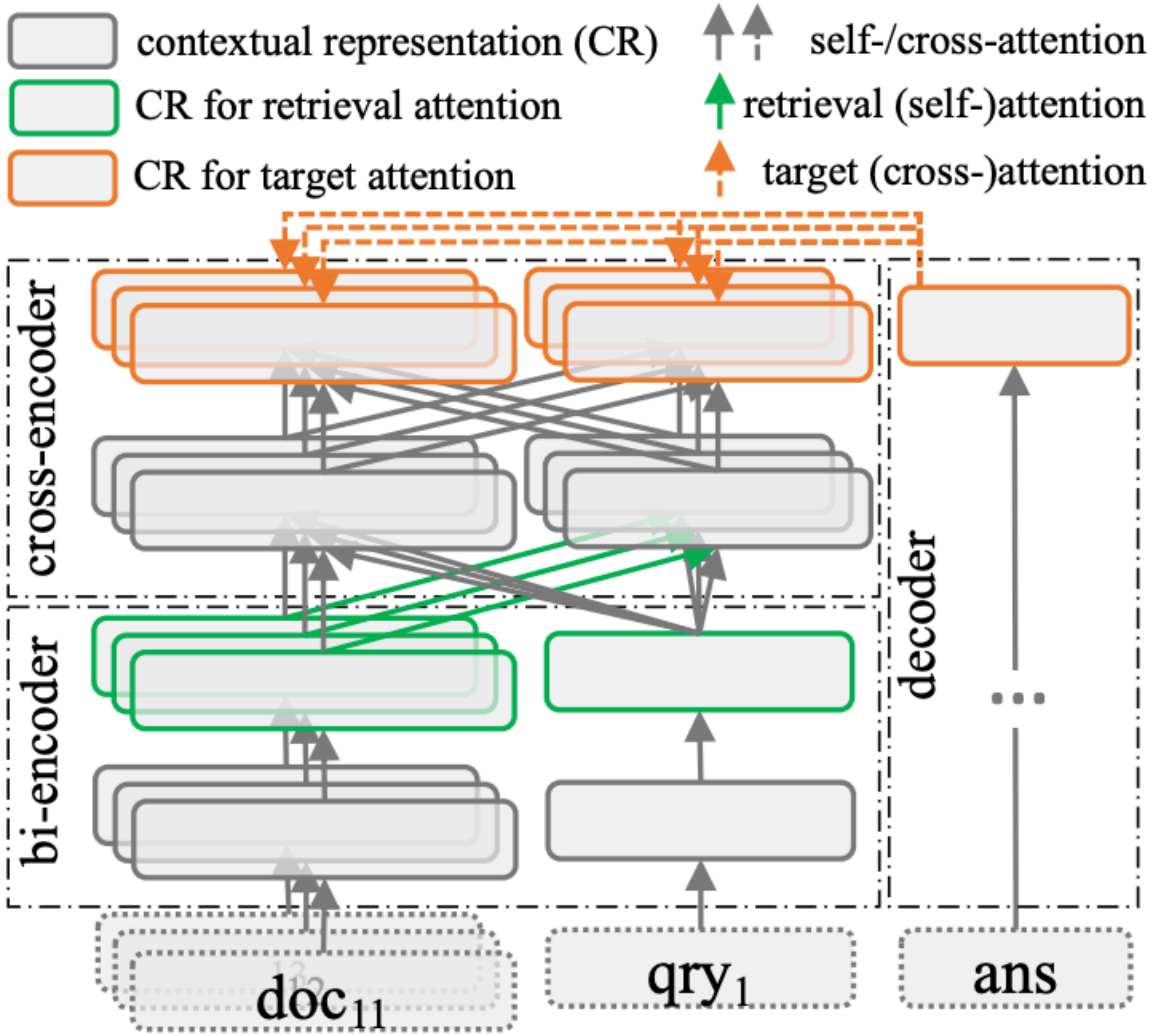
GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints - https://arxiv.org/pdf/2305.13245v3
Fast Transformer Decoding: One Write-Head is All You Need - https://arxiv.org/abs/1911.02150

One Transformer, Two Jobs

Act as a retriever and as a cross-encoder reader - **all inside a single model**

Retrieval is just another attention head: the model retrieves and reads in one forward pass.

End-to-end supervision: no separate IR labels



Zhengbao Jiang et al., EMNLP 2022

Memory: A Retrieval Problem

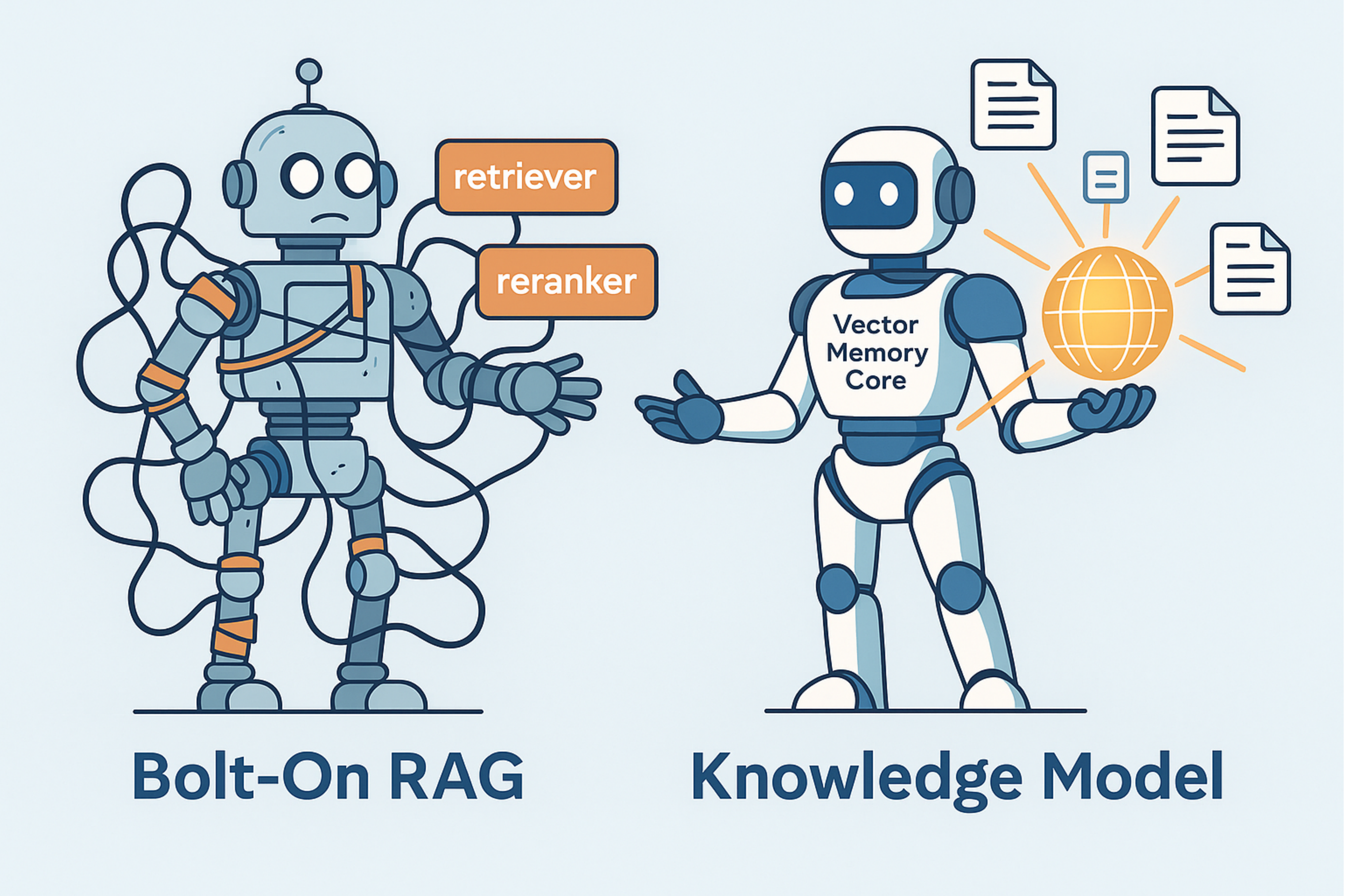
Retrieval = Attention. Embed large-scale retrieval inside the Transformer stack.

Built-in memory layer. The external datastore becomes an intrinsic component of the network..

Trillions-parameter capability, without trillion-parameter cost.

End-to-end trainable. Gradients flow through memory.

Rich research agenda: multi-query & group attention for paragraph-level keys, dynamic memory refreshing, diversified retrieval, and agentic “think-retrieve-think” loops that refine knowledge mid-generation.



Thanks!

Any questions?