

RAG in the Era of LLMs

From Basics to Industry Applications

Outline

- Introduction (5 min)
 - LLMs Briefly (15 min)
 - Limits of LLMs & Hallucination (10 min)
 - Introducing RAG (20 min)
 - Going Beyond Basic RAG (15 min)
 - Industry Applications (10 min)
 - Evaluation, Limitations & Future (10 min)
-
- **bad news** , you won't be able to build rag after this workshop **directly**
 - **good news** you will have a knowledge base that make you build things beyond simple RAG !

Introduction

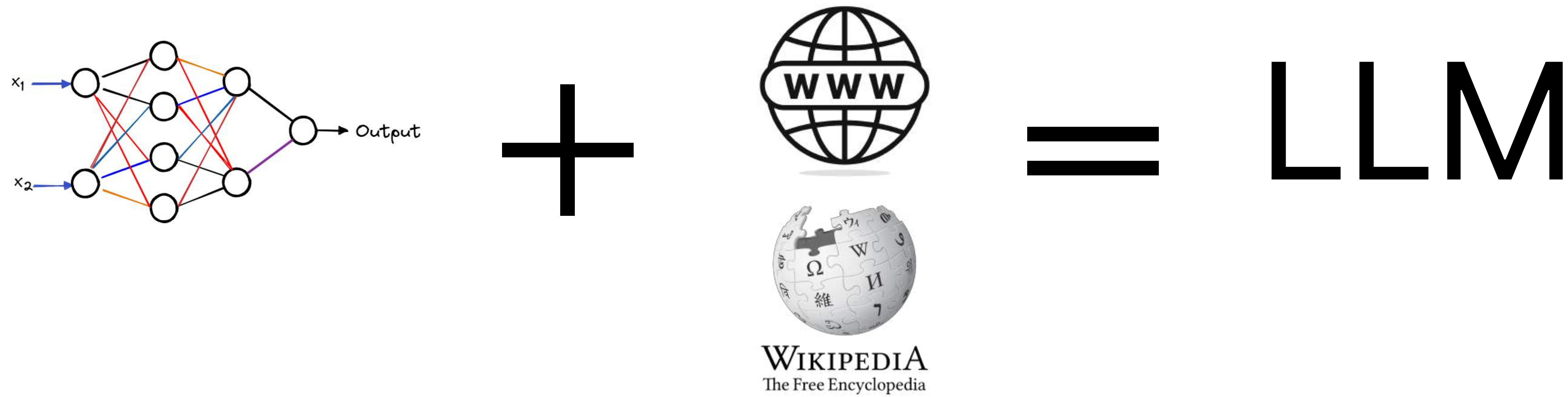
Motivation

- “LLMs revolutionized AI—but they hallucinate and forget.”
- “Industry needs domain-aware, up-to-date systems.”
- Question on slide: How can we make LLMs reliable and factual?



shutterstock.com · 2428260655

LLMs Briefly (15 min)



LLMs Briefly

A language model is a probabilistic model that give probabilities to sequence of words.
In practice, a language model allows us to compute :

$$P(\text{"Paris"} \mid \text{"What is the capital of France"})$$

if a language model is trained on large corpora , we called it a **Large Language Model**

OpenAI wasn't the first to introduce Language model , they exist from the **90s**

LLMs Briefly

Natural Language Processing With Modular PDP Networks and Distributed Lexicon ^{*†‡}

Risto Miikkulainen and Michael G. Dyer
University of California, Los Angeles

Abstract

An approach to connectionist natural language processing is proposed, which is based on hierarchically organized modular Parallel Distributed Processing (PDP) networks and a central lexicon of distributed input/output representations. The modules communicate using these representations, which are global and publicly available in the system. The representations are developed automat-

1991

LLMs Briefly

What Chatgpt Added

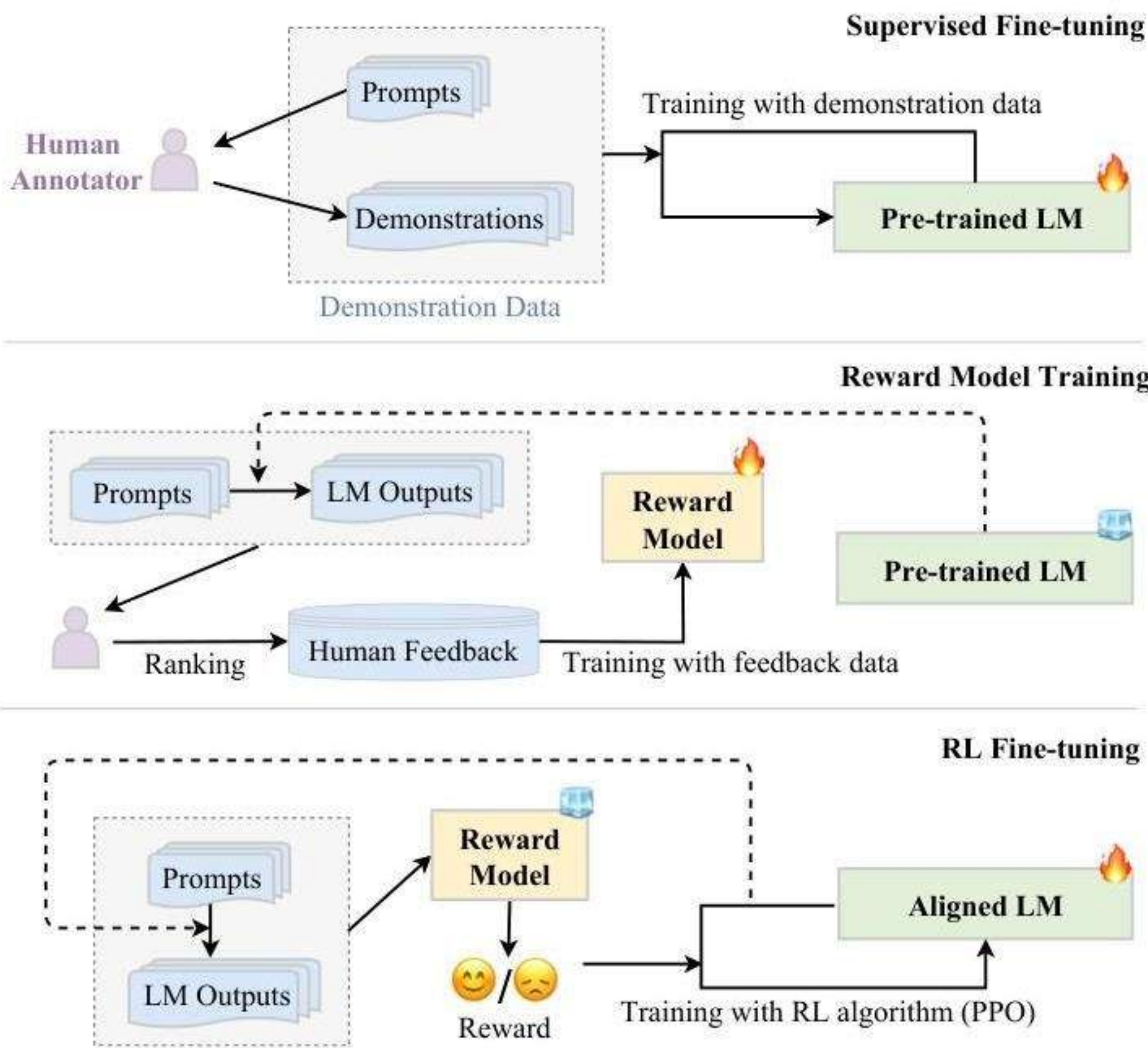
ChatGPT's innovation = user interface + instruction tuning + RLHF.

RAW LLMs aren't trained to follow instructions yet making them unusable by humans

LLMs Briefly

What Chatgpt Added

Instruction Tuning



LLM Survey, 2025

LLMs Briefly

What Chatgpt Added

Instruction Tuning

prompt string · lengths 35+27676.1%	completion string · lengths 1+31269.4%
I need you to write a resignation letter to my boss. My name: Anthony Company Name...	Hi Albert, Please accept this letter as formal notice that I'm giving my notice of resignation...
Joe Biden is the Nth president of the United States. What is N?	46
Write a four-sentence horror story about sleeping.	I woke up at 7 am after having nightmares the night before. As usual, I checked my phone as soon as my eyes opened. I felt a chill creeping up my spine, and saw that my wallpaper had been set to a picture of myself sleeping. Because I live alone.

LLMs Briefly

What Chatgpt Added

RLHF

teach me about RAG

Assistant A

Tools

LangChain, Haystack, Hugging Face

Data Needed

External knowledge base (corpus)

Output

More factual, up-to-date responses

Would you like to:

Try a real example?

See code in Python using LangChain or Hugging Face?

Learn how to implement a RAG chatbot?

Let me know!

Assistant B

Question answering systems

Chatbots with domain-specific knowledge

Document summarization

Research assistance

Example Implementation

A simple RAG system might use:

Vector database (like FAISS, Pinecone, or Chroma) for document storage and retrieval

Embedding model (like OpenAI's text-embedding-ada-002) to convert text to vectors

LLM (like GPT-4) for generation

Would you like me to explain any part of RAG in more detail or show you how to implement a basic RAG system?

← Left is Better

It's a tie

Both are bad

Right is Better →

10

LLMs Briefly

LLMs are interesting because of **scaling laws** and their consequences :

Emergent Abilities

LLMs Briefly

LLMs are interesting because of scaling laws and their consequences :

In context Learning

allows models to adapt to new tasks simply by including relevant examples in the input prompt.

LLMs Briefly

LLMs are interesting because of scaling laws and their consequences :

Chain-of-Thought Reasoning

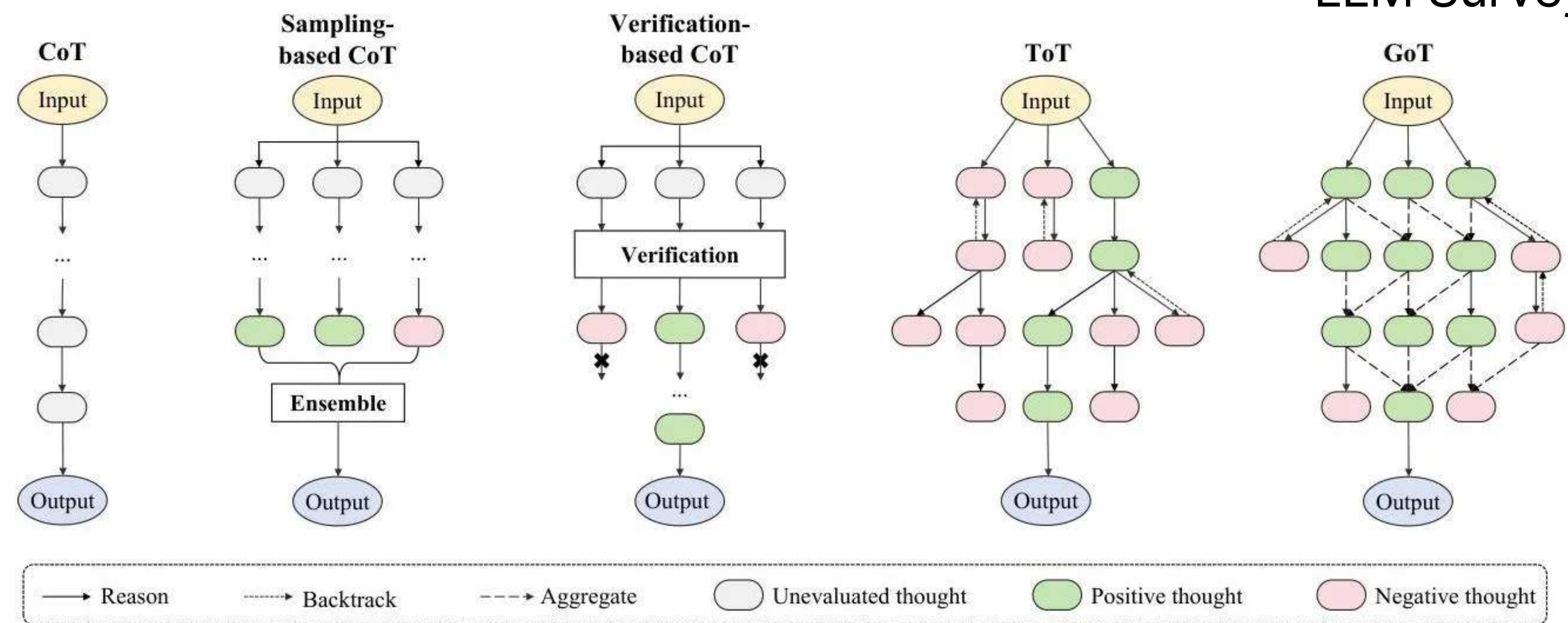
By encouraging models to break down problems into step-by-step reasoning processes, CoT significantly improves performance on mathematical, logical, and commonsense reasoning tasks.

LLMs Briefly

LLMs are interesting because of scaling laws and their consequences :

Chain-of-Thought Reasoning

LLM Survey, 2025



LLMs Briefly

LLMs are interesting because of scaling laws and their consequences :

A proposition

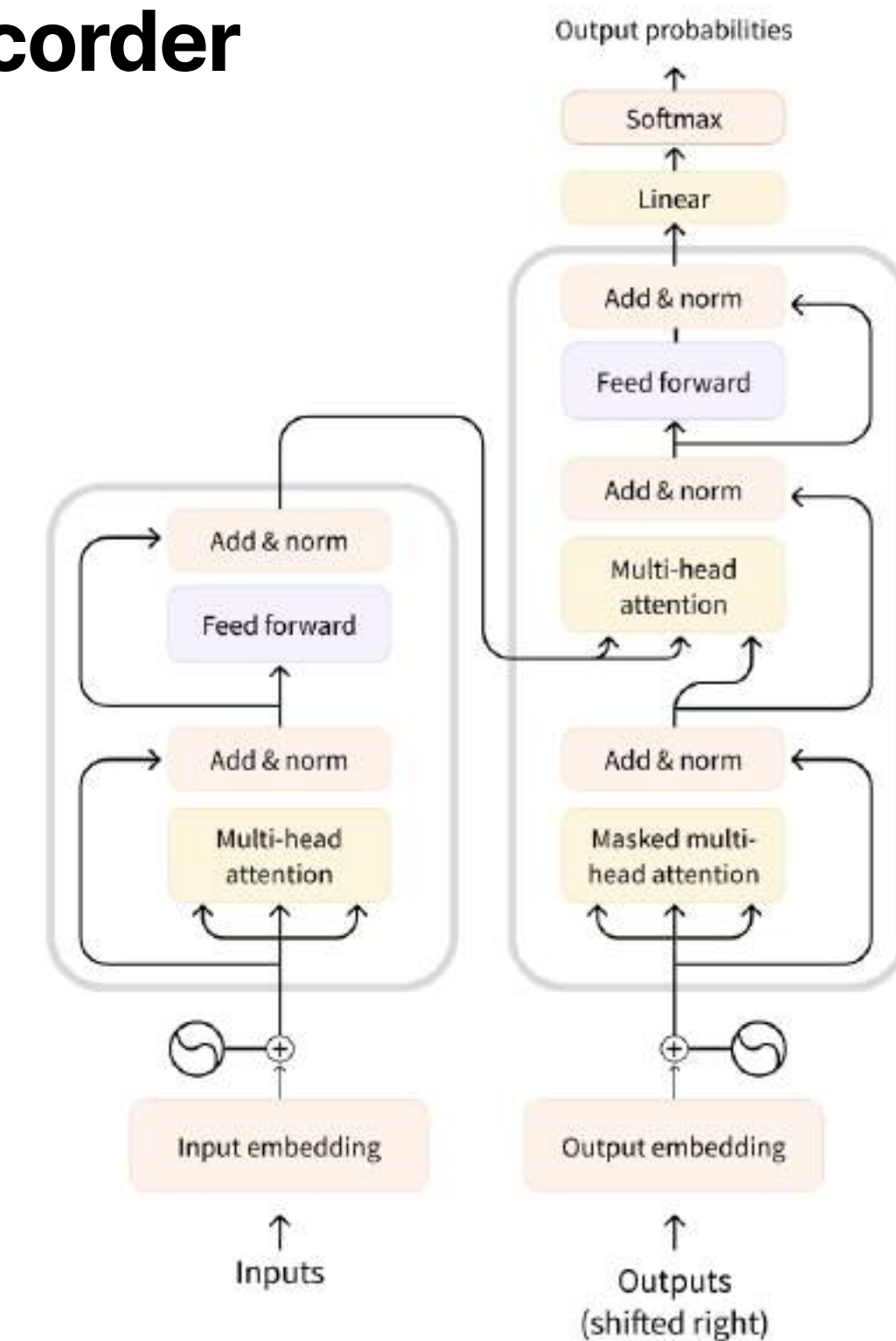
“If LLMs can reason for an infinite amount of tokens , they can solve any problem”

LLMs Briefly

- BERT (Encoder-only): Great for understanding context.
- GPT Series (Decoder-only): Excellent for text generation, follow instructions.
- T5 / BART (Encoder-Decoder): Versatile for tasks requiring understanding and generation (summarization, translation).

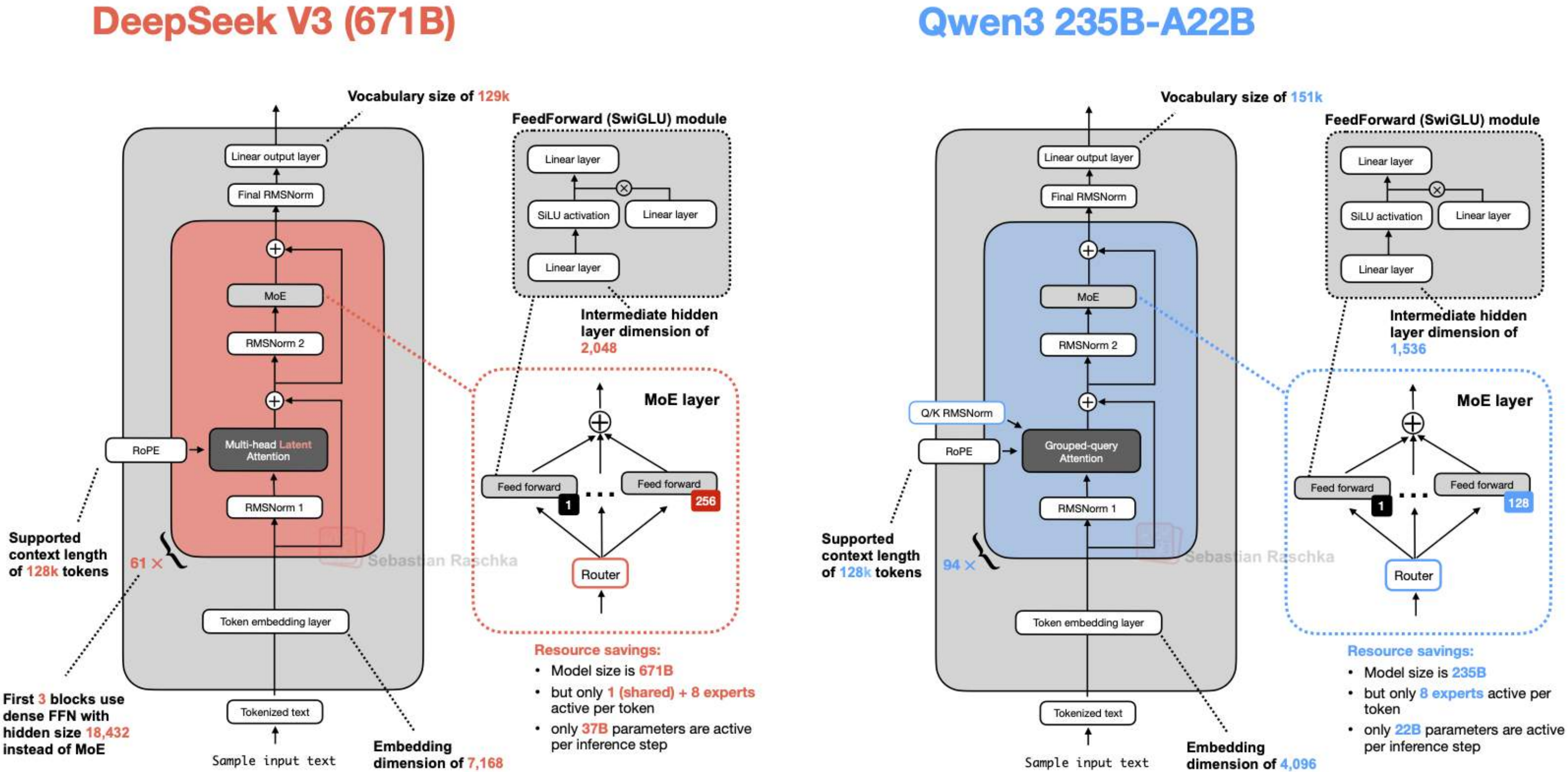


Encoder



Decoder

LLMs Briefly (15 min)



LLMs Briefly

A comprehensive analysis of "The Stack" v2 dataset (nearly 600 million code samples) revealed:

- **17% of code versions have newer versions available, indicating outdated code**
- **Of those with newer versions, 17% represent bug fixes**
- **2.36% of fixes address known CVE vulnerabilities**
- **The cleaned, deduplicated Stack v2 still includes code vulnerable to 6,947 known CVEs**
- **58% of code blobs were never modified after creation, suggesting minimal or no real-world usage**

Limits of LLMs & Hallucination

Limitations of Simple Input/Output

Garbage in Garbage Out : Prompt engineering is important

IV IVAN SEVERINO

What is a prompt?

CHATGPT

A prompt is a message or instruction given to elicit a response or action. In computing, it often refers to a symbol or set of symbols displayed in a command-line interface, indicating that the system is ready to accept input. In a broader sense, a prompt can also refer to any stimulus that seeks to initiate a reaction or feedback, like a question in a survey or a cue in psychological experiments.

IV IVAN SEVERINO

Please explain what is a prompt, like if I'm 5 years old

CHATGPT

Alright! Imagine you have a toy robot. If you ask it to dance, it dances. The question "Can you dance?" is like giving the robot a little nudge or hint about what to do next. That nudge or hint is called a "prompt." So, a prompt is like a little push or question you give to something (or someone) to get them to do or say something. 😊

Idea of writing in a paper a definition of rag randomly in the workshop , then simulating open book vs closed book . giving one attendee the paper and another not and asking them the same question

Limits of LLMs & Hallucination

Limitations of Simple Input/Output

Knowledge Gaps

Tied to its Training Data



Limits of LLMs & Hallucination

Limitations of Simple Input/Output

Reliability: Prone to hallucination without grounding or specific guidance.



How many 'm's are in the word 'Weather'?



There is one 'm' in the word 'Weather'.



Are you sure?



Apologies for the oversight. You're right; there are no 'm's in the word 'Weather'. Thank you for pointing it out.

Limits of LLMs & Hallucination

Why LLMs Fail Sometimes

- What do you really consider as hallucination ?
it's not about correctness , the model always predicts things based on the probability of them being correct , it's all about what the model considers being true not aligning with what we think is actually true (ground truth)!
- temperature doesn't solve the problem of hallucination!!!!!!

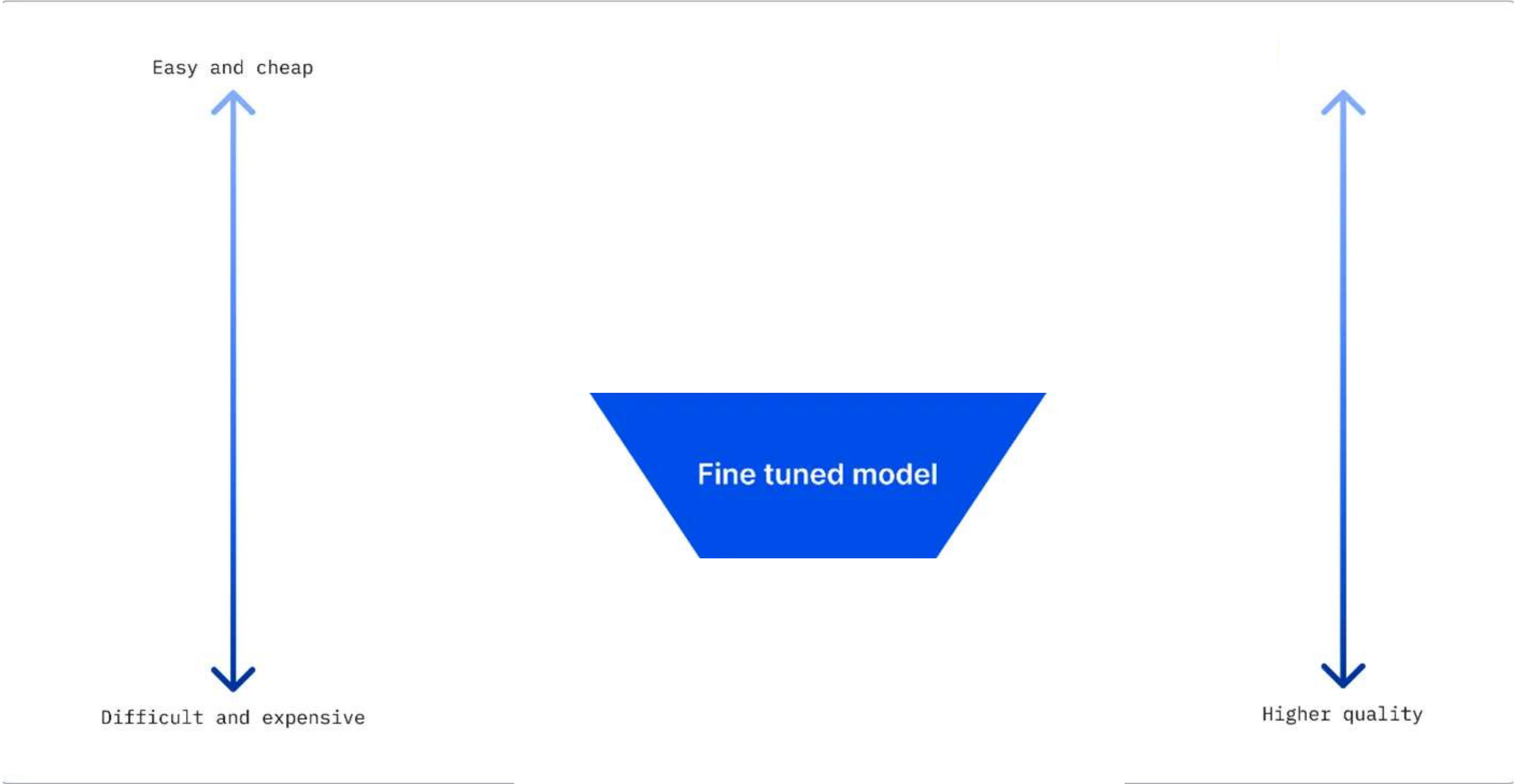
Limits of LLMs & Hallucination

You are what you eat

- A language model can only output text and information that it was trained upon. This means, that if we train a language model only on English content, very probably it won't be able to output Arabic or Kabyle.

To teach new concepts, we need to fine-tune the model.

Solving Hallucination



Solving Hallucination

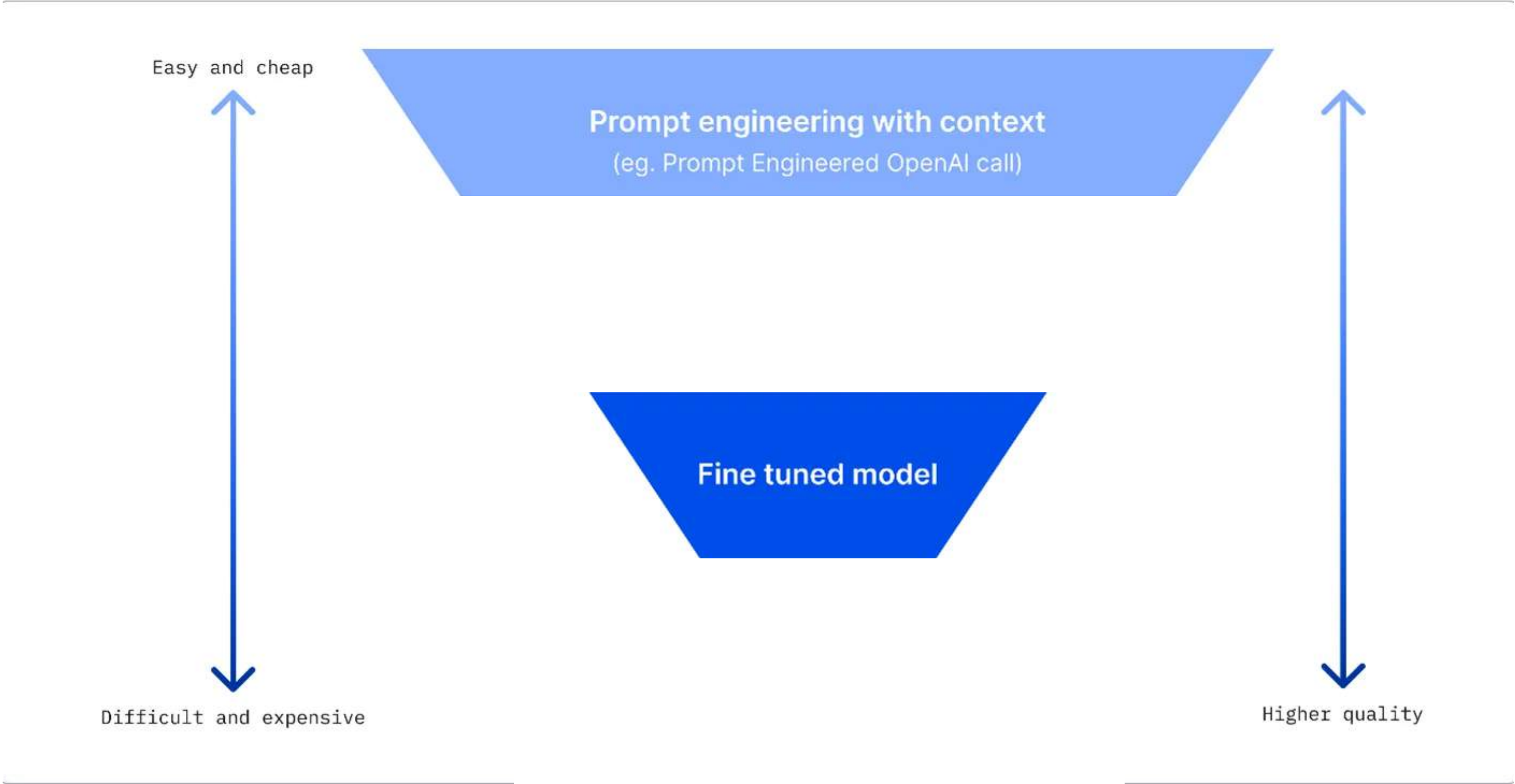
It can be expensive.

- The number of parameters of the model may not be sufficient to capture all the knowledge we want to teach to it. That's why LLaMA was introduced with 7B, 13B and 70B parameters.
- Fine-Tuning is not additive. It may replace existing knowledge of the model with new knowledge. For example, a language model trained on English that is (heavily) fine-tuned on Japanese may “forget” English.

Solving Hallucination

Prompt Engineering to the rescue!!

Teach without updating weights.



Solving Hallucination

Prompt Engineering (PE)

Crafting effective prompts to control LLM behavior:

- **System Prompts:** Define the LLM’s overall goal, persona, and constraints.
- **Task Decomposition:** Prompts guide the LLM to break down complex goals (e.g., "Think step-by-step...").
- **Output Formatting:** Prompts tell the LLM how to format output to match specific specific tool format.

In context learning Ability !!!

Goal	Act as travel guide and provide 3 travel suggestions		
Model	gemini-pro		
Temperature	1	Token Limit	1024
Top-K	40	Top-P	0.8
Prompt	I want you to act as a travel guide. I will write to you about my location and you will suggest 3 places to visit near me. In some cases, I will also give you the type of places I will visit. My suggestion: "I am in Amsterdam and I want to visit only museums." Travel Suggestions:		
Output	1. Rijksmuseum: World-renowned museum, known for Dutch masterpieces like Rembrandt’s “The Night Watch.” Don’t miss the incredible collection of paintings and artifacts showcasing Dutch art and history. 2. Van Gogh Museum: Immerse yourself in the captivating works of Vincent van Gogh. The museum houses the largest collection of his paintings and drawings, including “The Starry Night” and “Sunflowers.” 3. Stedelijk Museum Amsterdam: Discover modern and contemporary art from around the world. The museum’s collection includes works by Picasso, Kandinsky, and Rietveld, housed in a striking modern building.		

Table 5. An example of role prompting

Solving Hallucination

Prompt Engineering (PE)

In context learning Ability !!!

Crafting effective prompts to control LLM behavior:

- **System Prompts:** Define the LLM's overall goal, persona, and constraints.
- **Task Decomposition:** Prompts guide the LLM to break down complex goals (e.g., "Think step-by-step...").
- **Output Formatting:** Prompts tell the LLM how to format output to match specific specific tool format.

Prompt	Q: When my brother was 2 years old, I was double his age. Now I am 40 years old. How old is my brother? Let's think step by step. A: When my brother was 2 years, I was $2 * 2 = 4$ years old. That's an age difference of 2 years and I am older. Now I am 40 years old, so my brother is $40 - 2 = 38$ years old. The answer is 38. Q: When I was 3 years old, my partner was 3 times my age. Now, I am 20 years old. How old is my partner? Let's think step by step. A:
Output	When I was 3 years old, my partner was $3 * 3 = 9$ years old. That's an age difference of 6 years and my partner is older. Now I am 20 years old, so my partner is $20 + 6 = 26$ years old. The answer is 26.

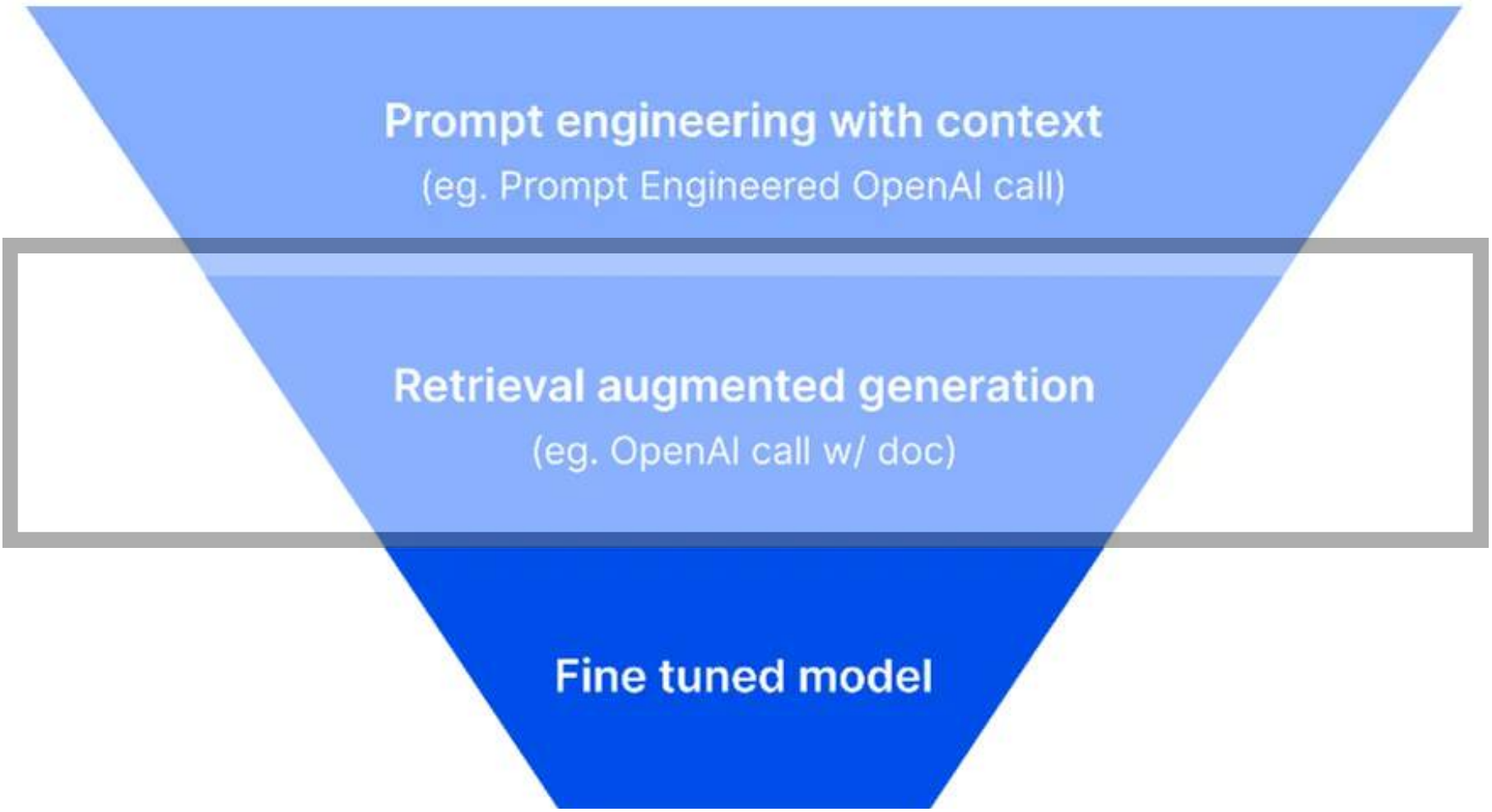
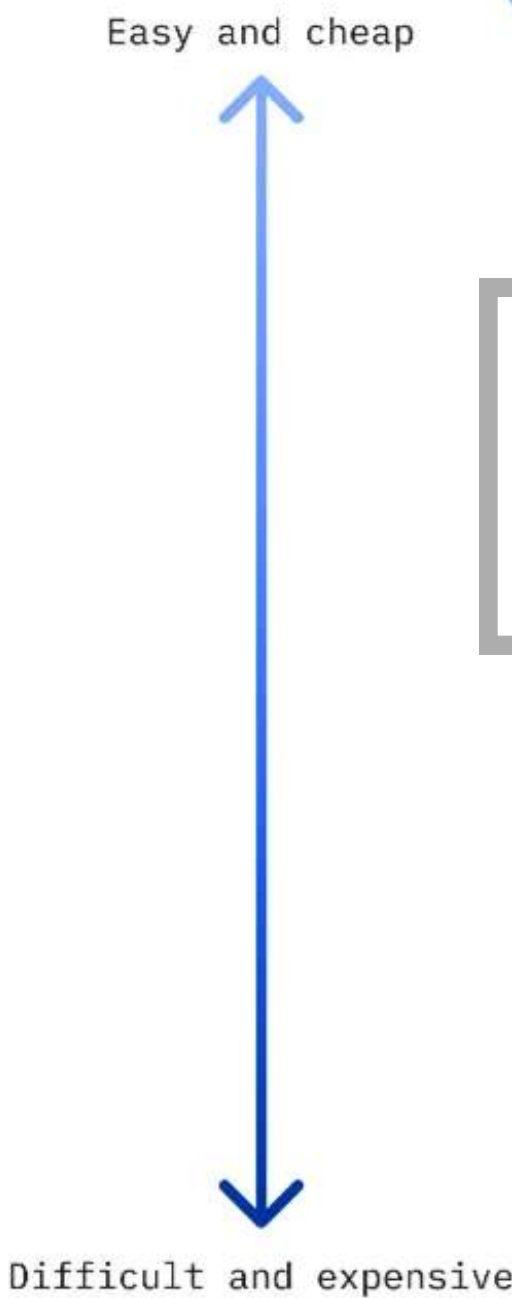
Table 13. An example of chain of thought prompting with a single-shot

Solving Hallucination

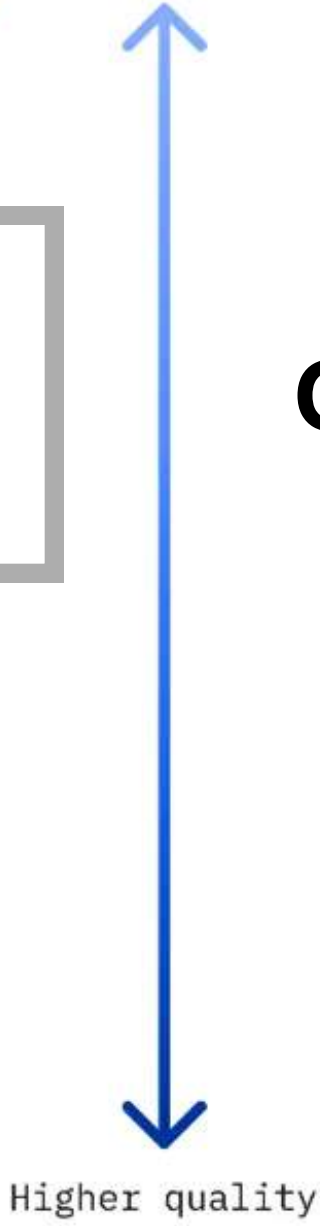
RAG to the rescue !

Fine-tuning teaches the model (training) — RAG gives it access to knowledge at inference time.

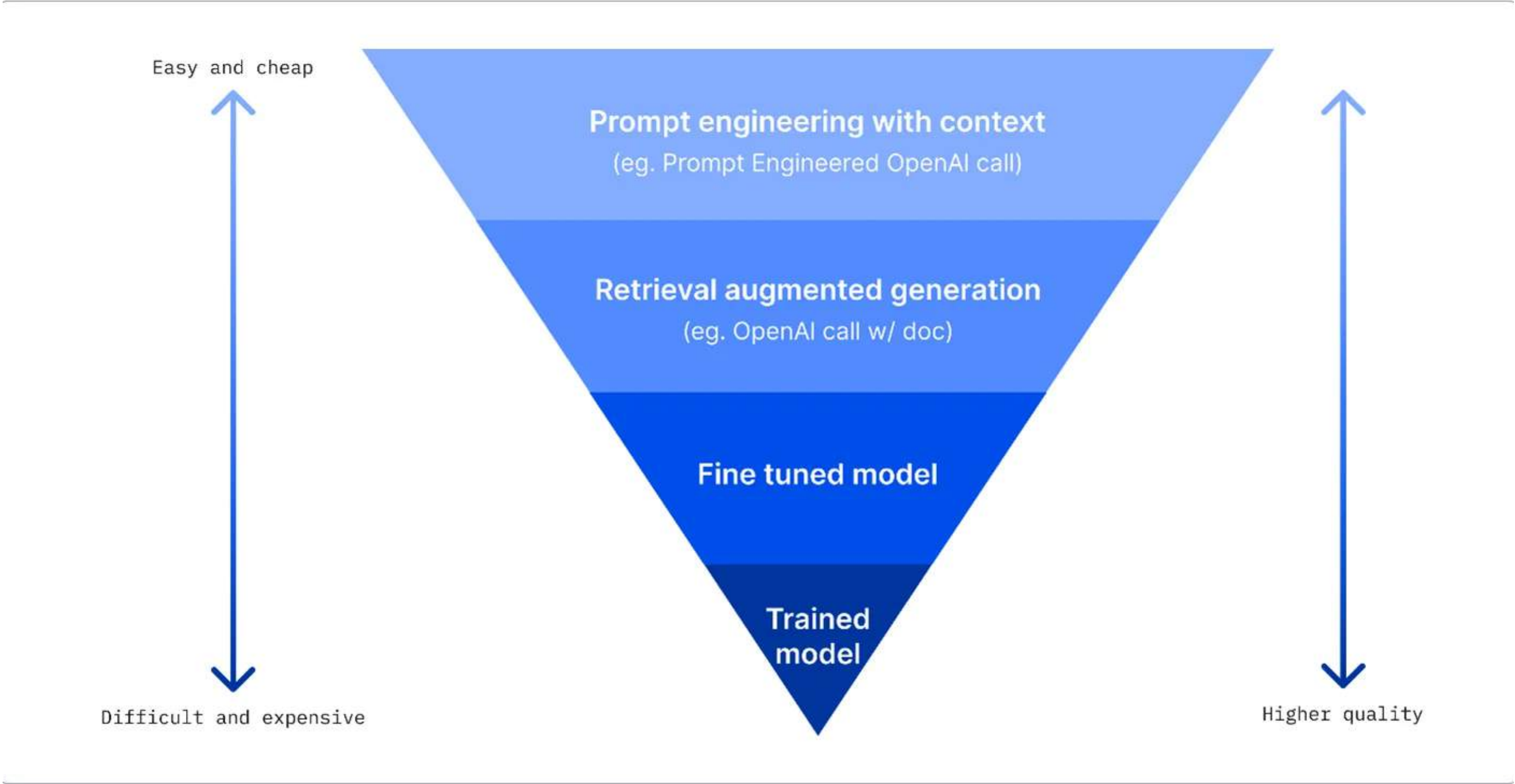
Closed Book



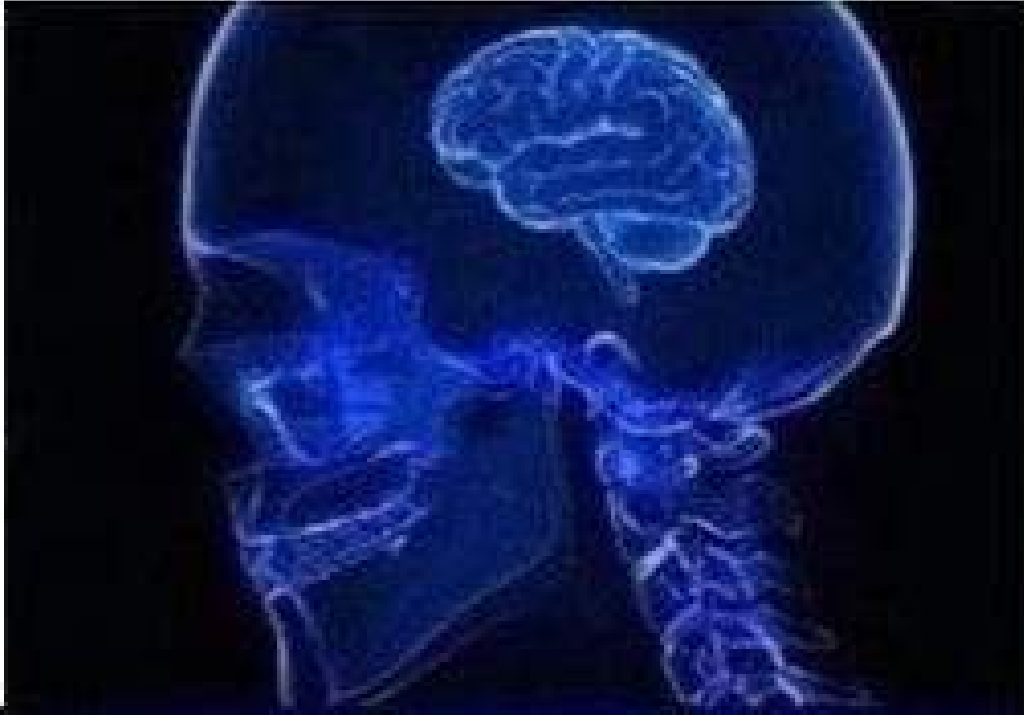
Open Book



Solving Hallucination



**PROMPT
ENGINEERING**



RAG



FINE-TUNING



PRE-TRAINING



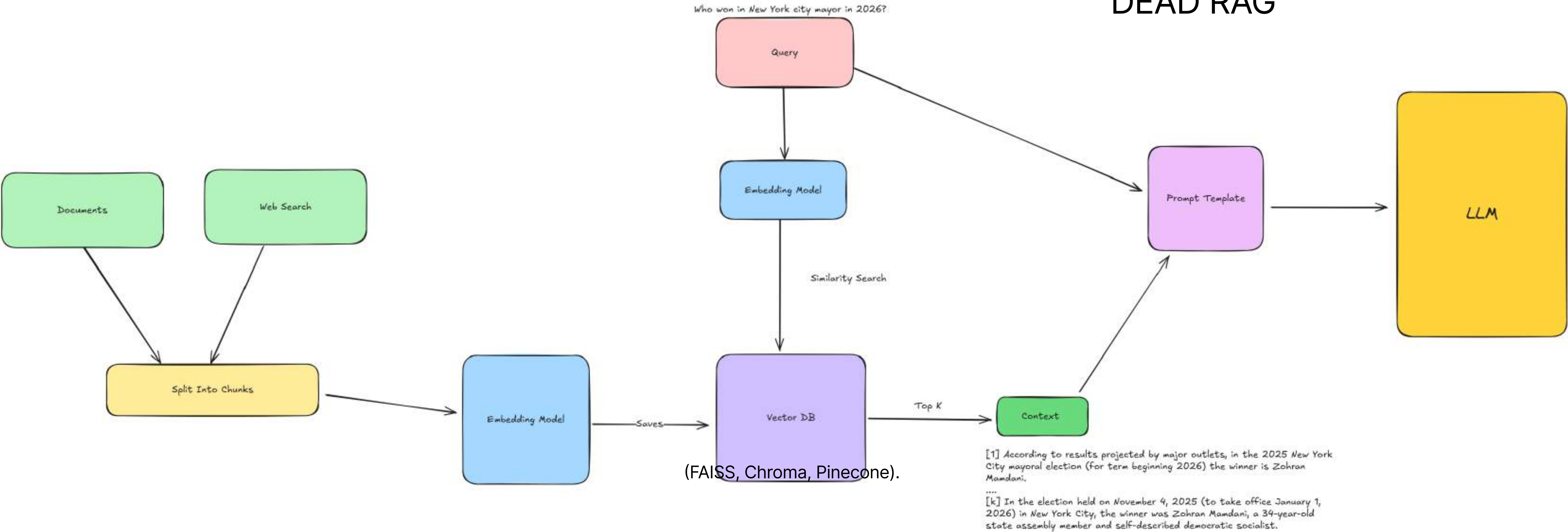
Introducing RAG

High Level , architecture

Frozen RAG

Naive RAG

DEAD RAG



Introducing RAG

High Level , architecture

https://colab.research.google.com/github/NirDiamant/RAG_Techniques/blob/main/all_rag_techniques/simple_rag.ipynb

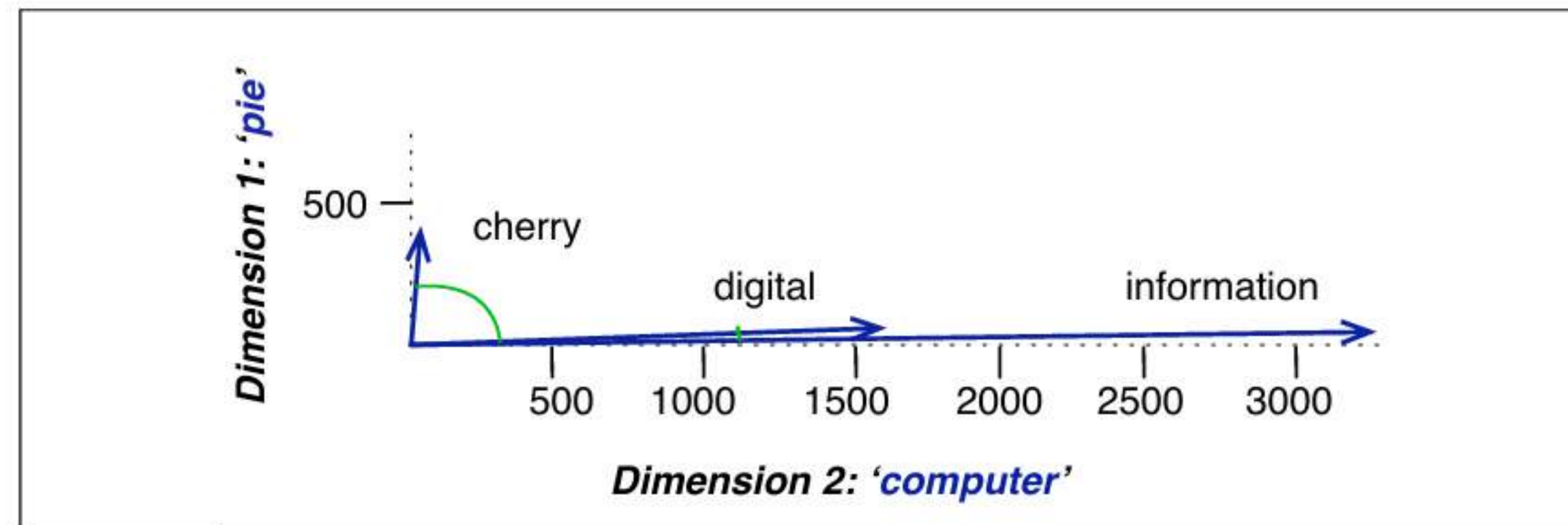
(FAISS, Chroma, Pinecone).

Introducing RAG

Why Embeddings & Vector Databases

Given the words “**cherry**”, “**digital**” and “**information**”, if we represent the embedding vectors using only **2 dimensions (X, Y)** and we plot them, we hope to see something like this: the angle between words with similar meaning is small, while the angle between words with different meaning is big. So, **the embeddings “capture” the meaning of the words they represent by projecting them into a high-dimensional space**

We commonly use the **cosine similarity**, which is based on the dot product between the two vectors



Source: Speech and Language Processing 3rd Edition Draft, Dan Jurafsky and James H. Martin

Introducing RAG

Why Embeddings & Vector Databases

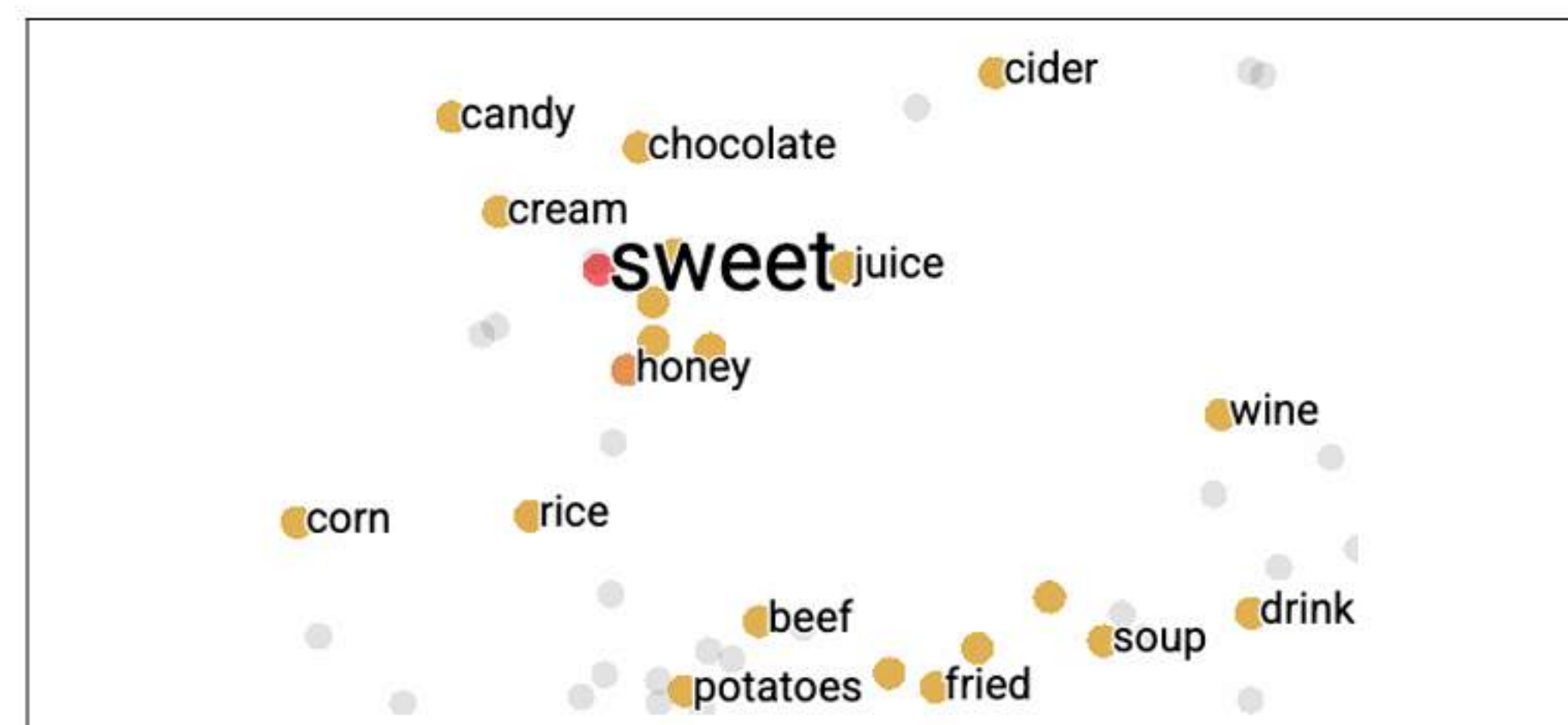
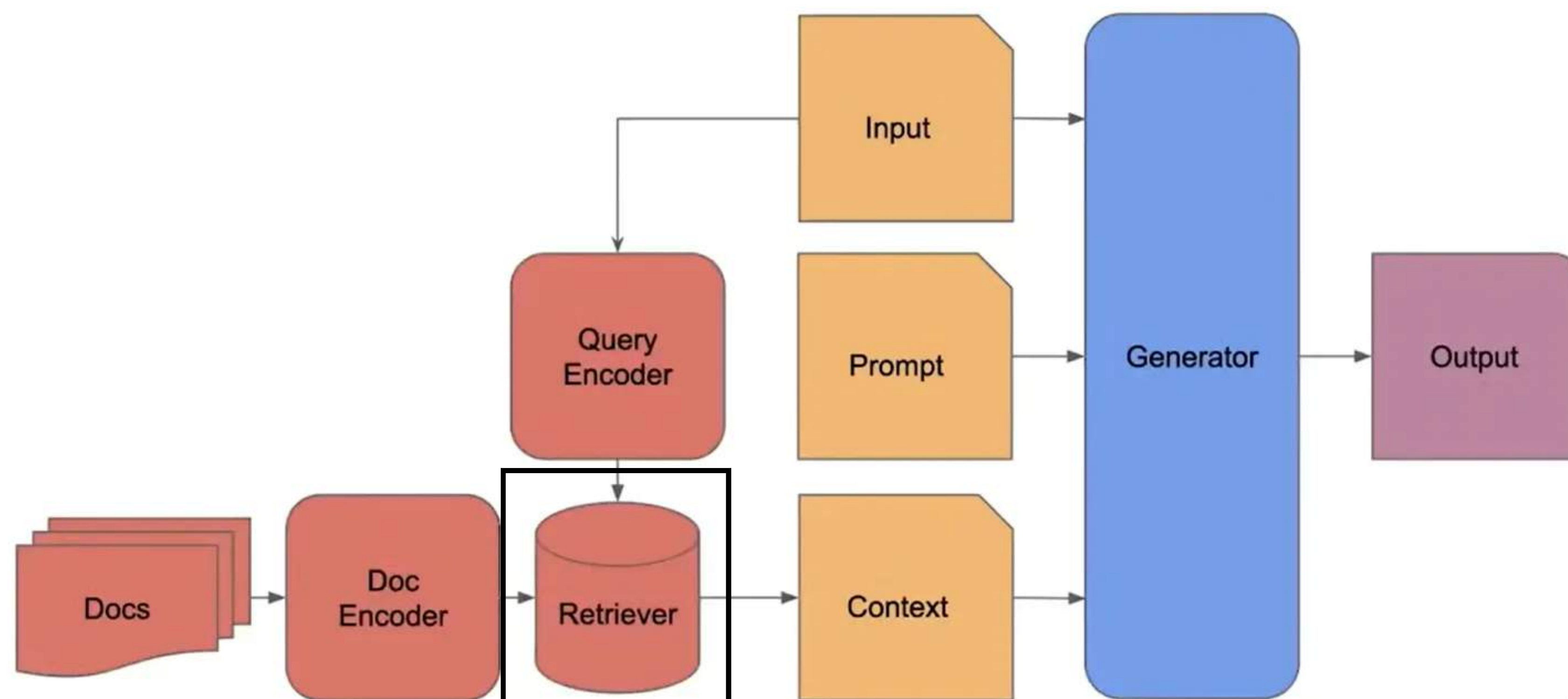


Figure 5.1 A two-dimensional (t-SNE) visualization of 200-dimensional word2vec embeddings for some words close to the word *sweet*, showing that words with similar meanings are nearby in space. Visualization created using the TensorBoard Embedding Projector <https://projector.tensorflow.org/>.

Source: Speech and Language Processing 3rd Edition Draft, Dan Jurafsky and James H. Martin

Going Beyond Basic RAG



Going Beyond Basic RAG

BM25 retrieval

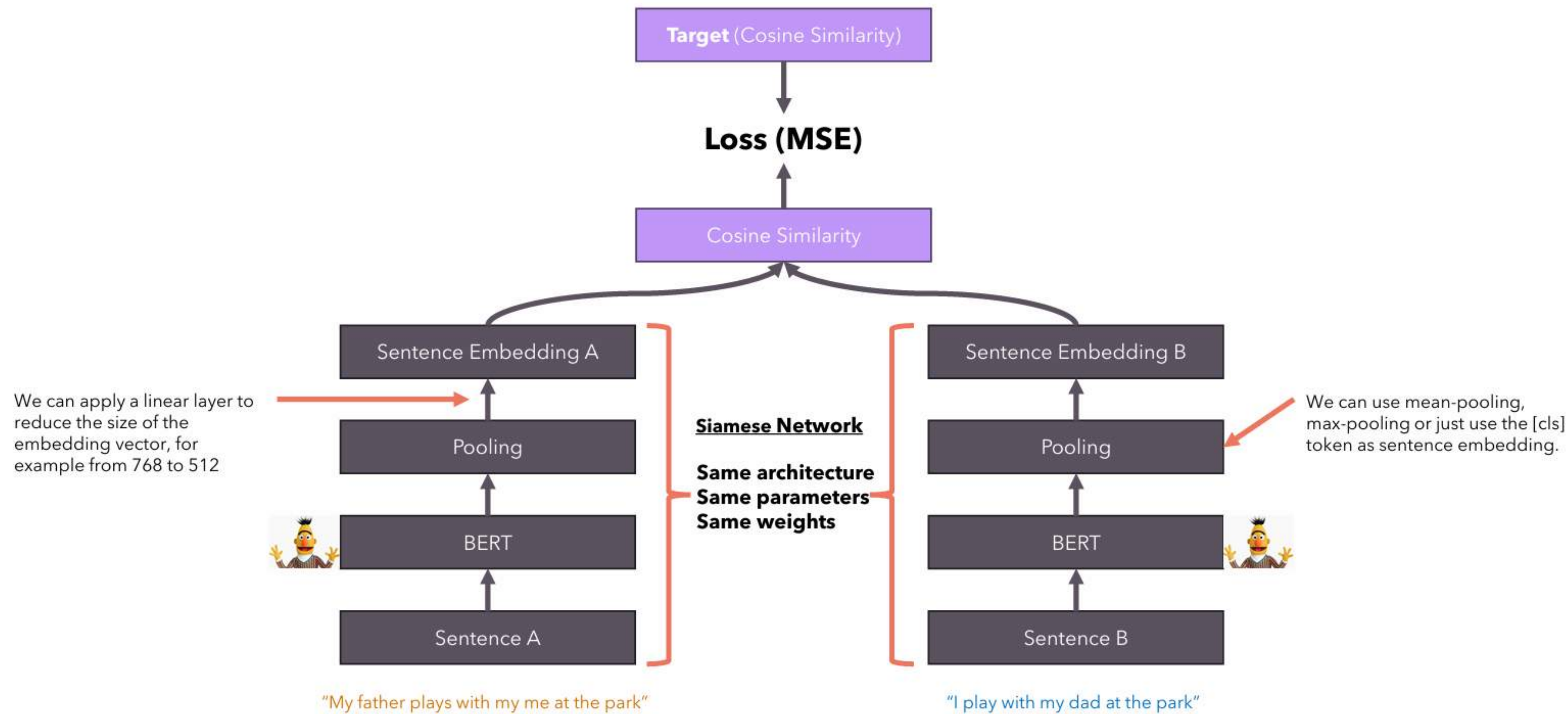
Enhanced version of TF-IDF

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Where $f(q_i, D)$ = frequency of term q_i in document D , $|D|$ = length of document, avgdl = average document length across the collection

Going Beyond Basic RAG

Dense Retrieval



Going Beyond Basic RAG

Sparse vs Dense Retrieval

Aspect	BM25 (Sparse/Term-based)	Dense/Embedding-based Retrieval
Representation	Term / lexical features (inverted index)	Dense vector embeddings (semantic features)
Semantic matching	Exact term or near-term matches	Captures synonyms, paraphrases, conceptual similarity
Computation cost	Low (inverted index lookups)	Higher (embedding generation, similarity search, GPU usage)
Interpretability	High — scoring formula transparent	Often lower — model internal weights less interpretable
Storage / indexing	Sparse index structure, efficient	Requires storing high-dimensional vectors, approximate nearest-neighbour (ANN) structures
Hybrid usage	Often used for first-stage retrieval	Often used for re-ranking or full retrieval in semantic tasks

Going Beyond Basic RAG

Sparse vs Dense Retrieval

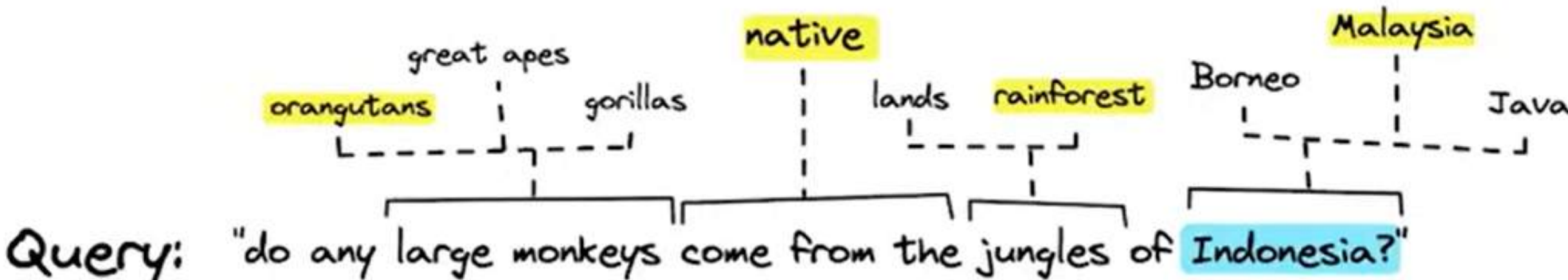
Generally they use both

Aspect	BM25 (Sparse/Term-based)	Dense/Embedding-based Retrieval
Representation	Term / lexical features (inverted index)	Dense vector embeddings (semantic features)
Semantic matching	Exact term or near-term matches	Captures synonyms, paraphrases, conceptual similarity
Computation cost	Low (inverted index lookups)	Higher (embedding generation, similarity search, GPU usage)
Interpretability	High — scoring formula transparent	Often lower — model internal weights less interpretable
Storage / indexing	Sparse index structure, efficient	Requires storing high-dimensional vectors, approximate nearest-neighbour (ANN) structures
Hybrid usage	Often used for first-stage retrieval	Often used for re-ranking or full retrieval in semantic tasks

Going Beyond Basic RAG

SOTA

- SPLADE:



Sparse meets dense with query expansion (Formal et al 2021).

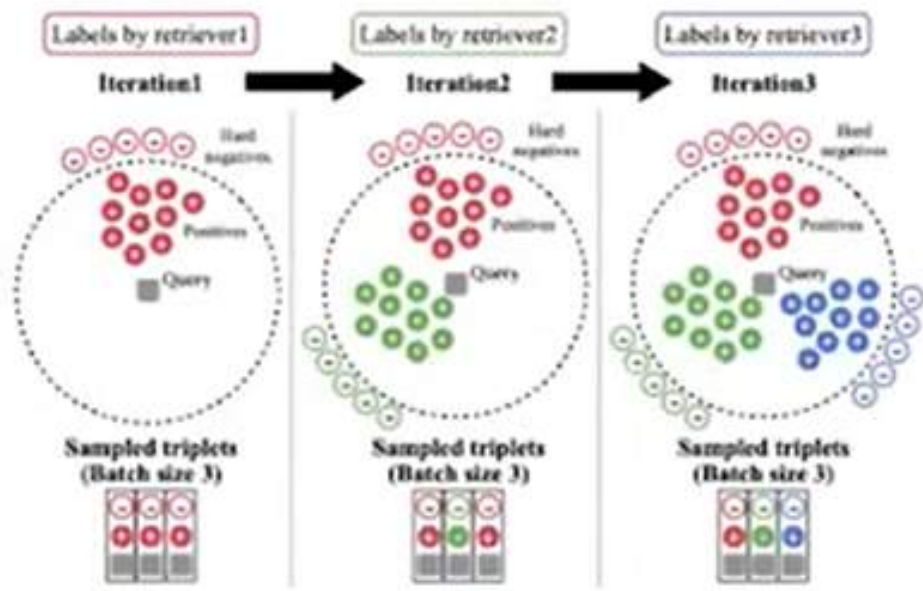
with query expansion

without query expansion

Doc: "Orangutans are native to the rainforests of Indonesia and Malaysia"

- DRAGON:

Generalized dense retriever via progressive data augmentation (Lin et al 2023).



- Hybrid search, where we interpolate between sparse and dense.

Going Beyond Basic RAG

REPLUG

RePlug: retriever optimized through feedback from the LLM itself.

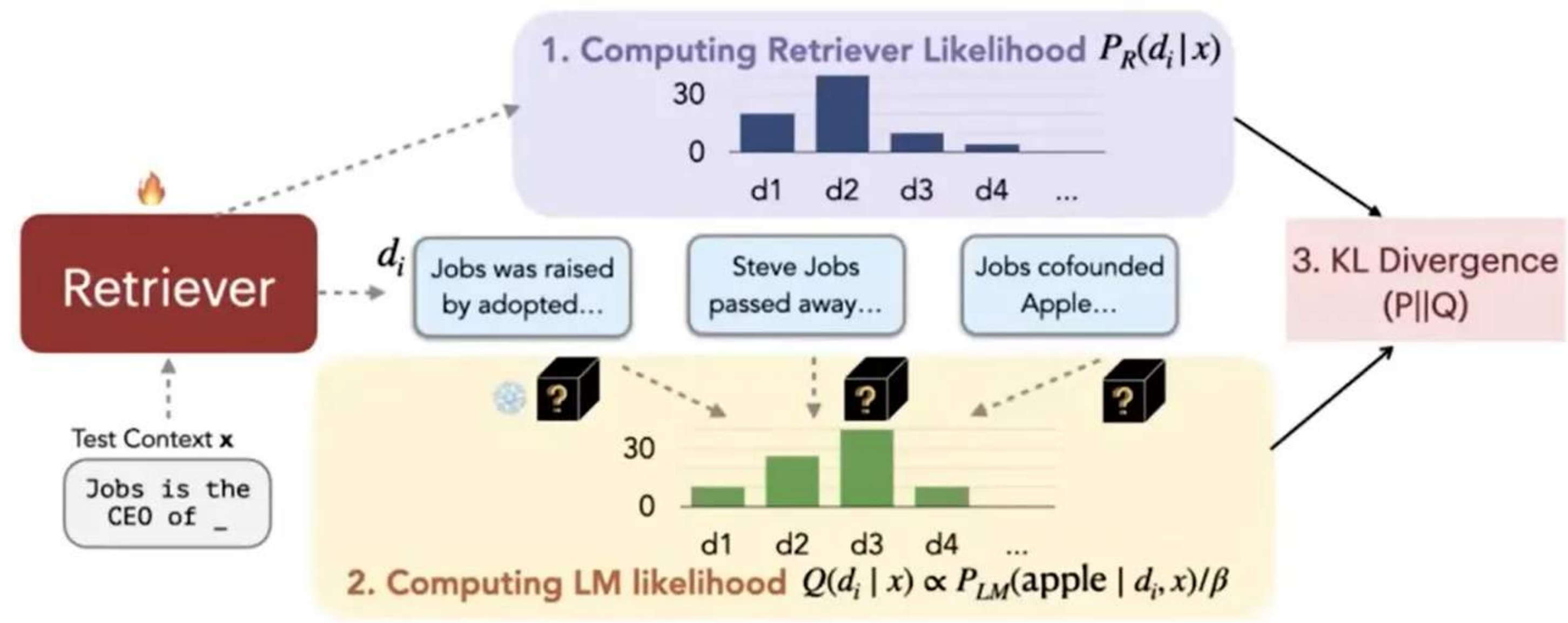
- Uses perplexity (confidence) to measure how useful retrieved documents are.
- Minimizes KL Divergence between retrieval distribution and model confidence.

Generator(LLM) remains frozen — retriever learns from generator's feedback.

The training objective is to **align these two distributions**: we want the retriever to pick documents that the LLM is confident in (i.e., high probability of correct output).

Going Beyond Basic RAG

RePlug (Shi et al 2023)



Self RAG

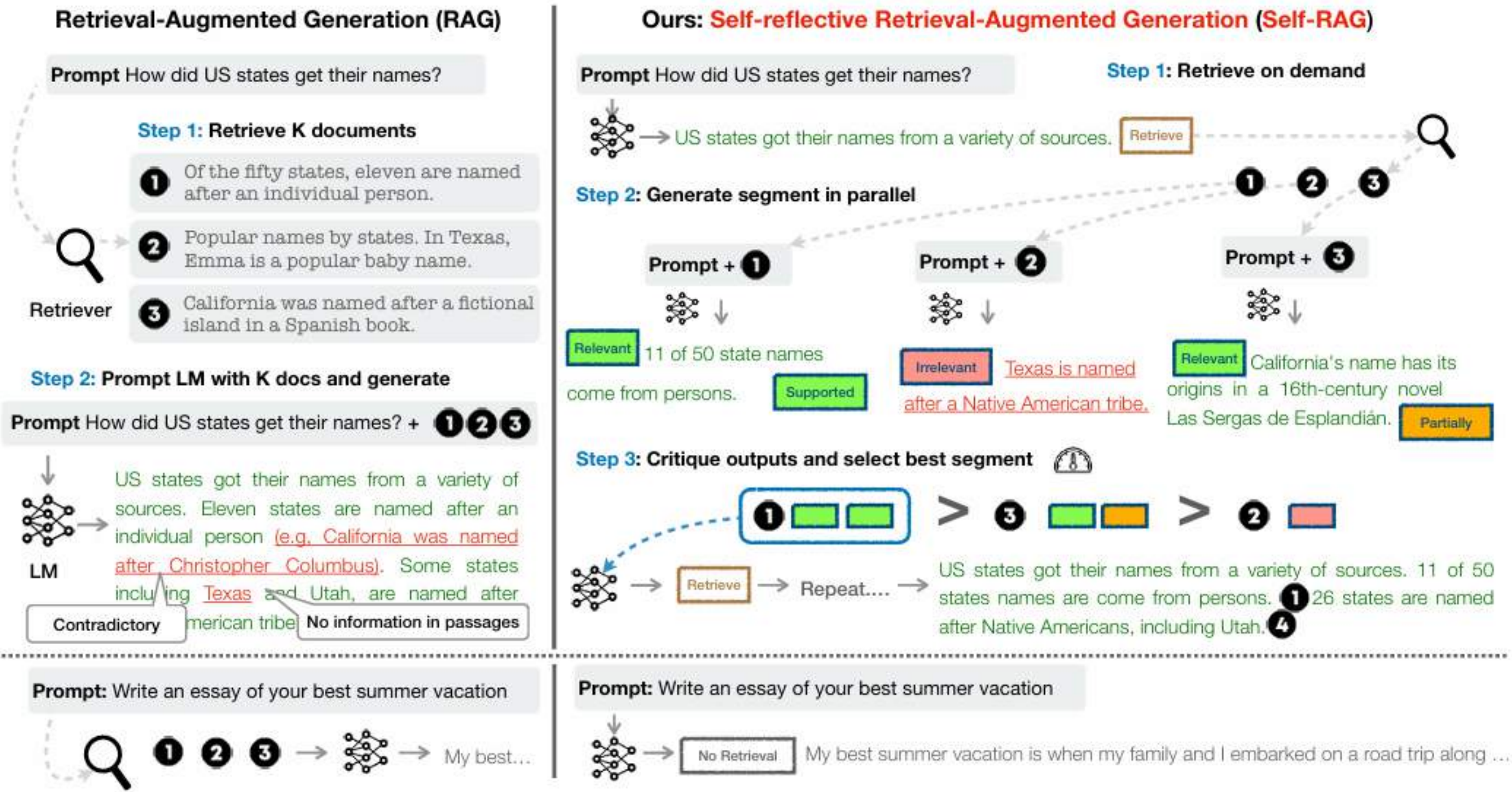


Figure 1: Overview of SELF-RAG. SELF-RAG learns to retrieve, critique, and generate text passages to enhance overall generation quality, factuality, and verifiability.

Self RAG

Self-RAG make the LLM act like **a student with a library**: it first decides if it needs to **consult books**, picks the best ones, writes an answer, then asks **‘Did I pick good books? Did I actually use them properly?’** If not, it goes back and retrieves again. This loop gives much better factual grounding and avoids retrieving blindly and using all chunks every time.

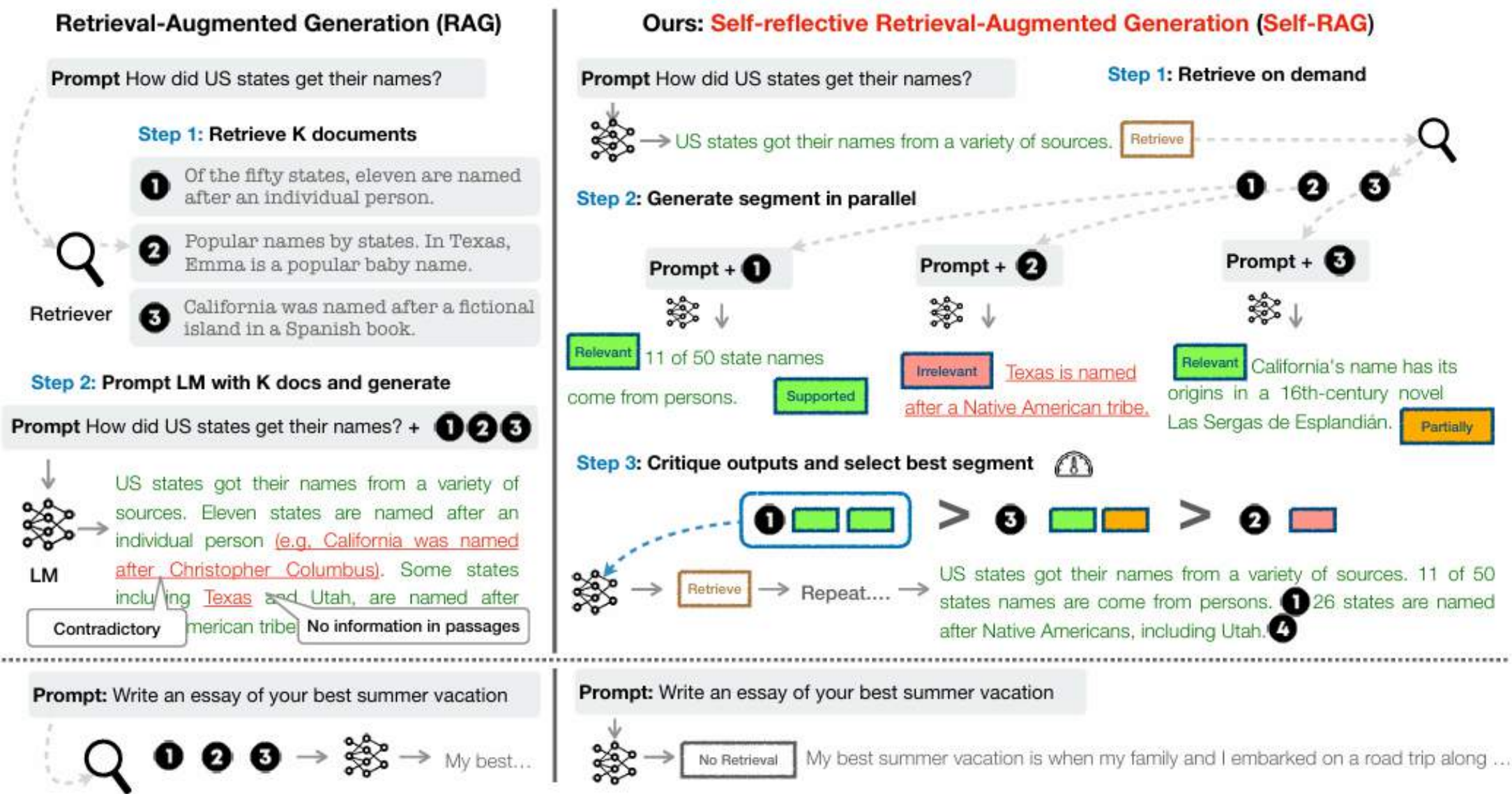
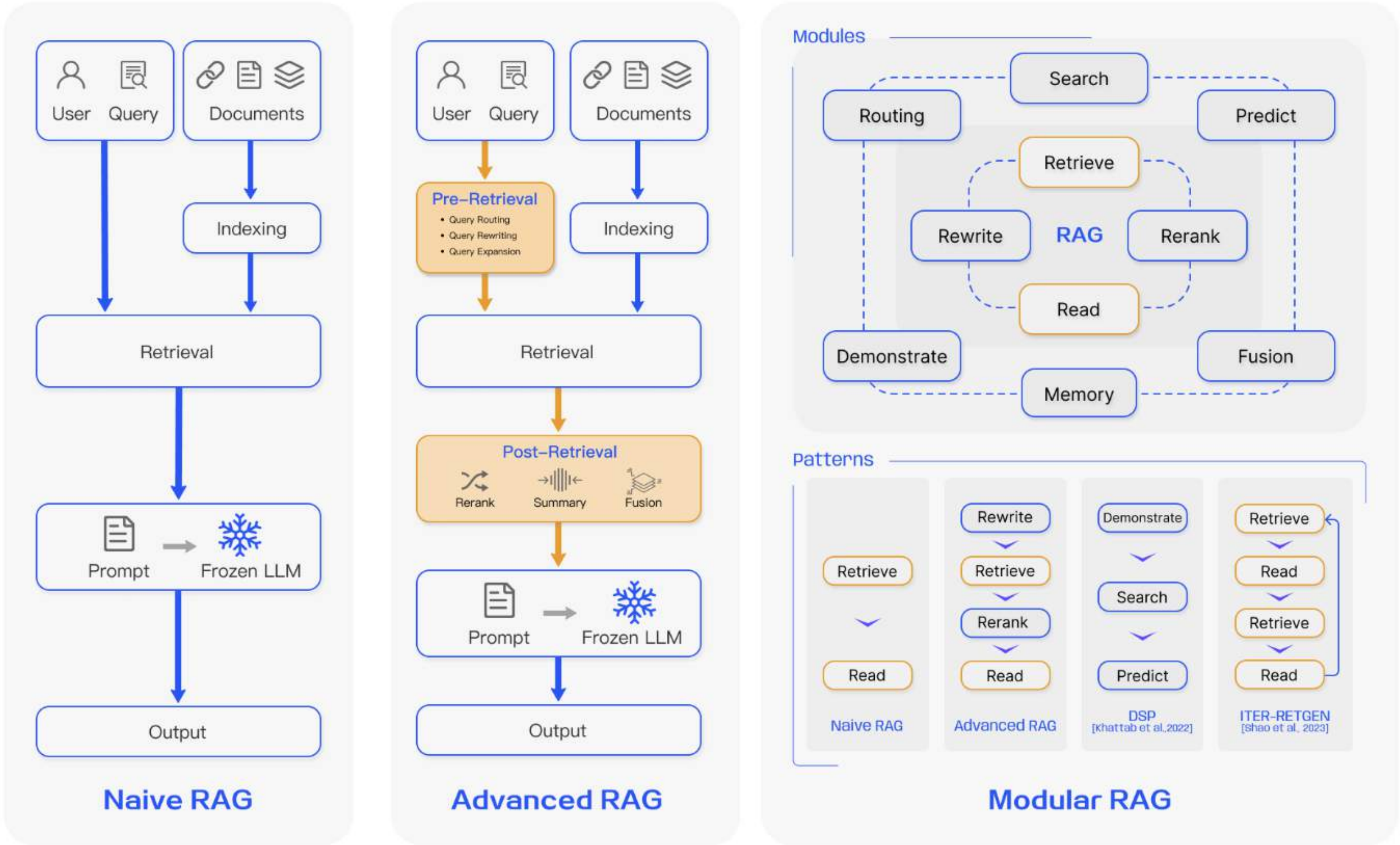


Figure 1: Overview of SELF-RAG. SELF-RAG learns to retrieve, critique, and generate text passages to enhance overall generation quality, factuality, and verifiability.

RAG 2.0



RAG 2.0

RAG 2.0:

- Retriever + Generator co-trained as one system.
- Moves from retrieval to reasoning synergy.

Context Order Matters:

- LLMs tend to prioritize first and last chunks.
- Smart chunk placement improves grounding.

RAG 2.0

Advanced RAG

- **1. Pre-Retrieval:** This involves optimizing the knowledge source before a search even occurs. Techniques include improving how data is indexed, adding metadata, and structuring the data for better alignment with potential queries.
- **2. Retrieval:** During the search, the system can be optimized by filtering results, for instance, by using a similarity threshold to discard irrelevant information and reduce the chances of generating misleading or false content.
- **3. Post-Retrieval:** After information is found but before it's sent to the LLM, this step refines the retrieved content. A key technique is to reorder the information to place the most relevant material first, helping the LLM focus on the most crucial context and work within its token limits.

RAG 2.0

Modular RAG

1. Specialized Modules: The system is built from components, each with a specific function.

An example given is a system with separate modules for:

- Hybrid Retrievers: Using multiple search methods at once to get the best results.
- Query Augmentation: A module that rewrites the user's initial question to be more effective for searching.
- LLM Reader: A module that provides specific medical context to the final question before the LLM answers it.

2. Integration of Advanced Techniques: Modular RAG isn't limited to just retrieval. It can be combined with other machine learning techniques. an example is framework that integrates RAG with the XGBoost algorithm to improve disease prediction, showing how it can become part of a larger, more complex system

RAG 2.0

- 3. Rethinking the Core Process:** This approach is so flexible it can even eliminate core components of traditional RAG. The example of "prompt-RAG" is significant because it gets rid of vector embeddings and databases entirely, instead using another LLM to handle the retrieval step with natural language prompts. This simplifies the process but, as noted, can be much slower.
- 4. Multi-LLM and Agentic Systems:** This is a major trend within Modular RAG. Instead of using a single LLM, researchers are building systems where multiple LLMs work together.
- **LLM Chains:** One LLM might perform a search, a second one might classify the results, a third summarizes them, and a fourth generates the final answer.
 - **Multi-Agent Systems:** This involves creating multiple specialized AI "agents" that can divide a complex problem into smaller sub-tasks, with each agent tackling a piece of the problem. This has been shown to dramatically improve accuracy.
 - **Fine-Tuning Specific Modules:** This involves adapting a specific component, such as the retriever, to a particular domain or task to improve its performance without having to retrain the entire system.

Industry Applications

-  Chatbots for internal documentation & enterprise QA
-  Legal assistants retrieving laws or precedents
-  Medical assistants with verified medical corpora
-  Product assistants searching manuals
-  Research tools for academic paper QA
-  Real-world examples:
 - Perplexity.ai
 - ChatGPT Web Search

Industry Applications

- Devfest23 Algiers has a chatbot that uses RAG and replies to questions about Previous Devfest Editions

Future Trends

-  Agentic RAG:
 - LLMs autonomously decide what, when, and how to retrieve.
-  Retrieval + Reasoning Co-Training:
 - Joint optimization of retriever and generator.
-  Multi-Modal RAG:
 - Combine text, tables, and images in retrieval.
-  Memory & Multi-Agent Systems:
 - Long-term retrieval memory + collaboration between agents.

Frameworks



LangGraph



LlamaIndex



LangChain

And more ...



smolagents



Summary

- ◆ Fine-tuning changes the model's weights — RAG changes its context.
- ◆ RAG = cheaper, grounded, and continuously updatable.
- ◆ Build systems, not just models.
- ◆ Future = agentic, tool-using, reasoning-driven AI.

“Think of RAG as giving the model an evolving memory instead of a static brain. The real innovation isn't a bigger LLM — it's smarter integration.”

Conclusion

RAG = grounding knowledge dynamically.

Resources & Further Learning

- <https://langchain-ai.github.io/langgraph/tutorials/introduction/>
- <https://huggingface.co/learn/agents-course/unit0/introduction>
- <https://www.ibm.com/think/topics/parameter-efficient-fine-tuning>
- <https://www.linkedin.com/feed/update/urn:li:activity:7322560318465892352/>
- <https://github.com/Chamiln17/generative-ai-for-beginners/tree/main/01-introduction-to-genai>
- <https://www.geeksforgeeks.org/nlp/what-is-bm25-best-matching-25-algorithm/>
- <https://arxiv.org/pdf/2310.11511>
- <https://arxiv.org/pdf/2312.10997>

Q&A

