Retrieval Augmented Generation (RAG)

AI advancement and adaption

Madhu Chinnambeti

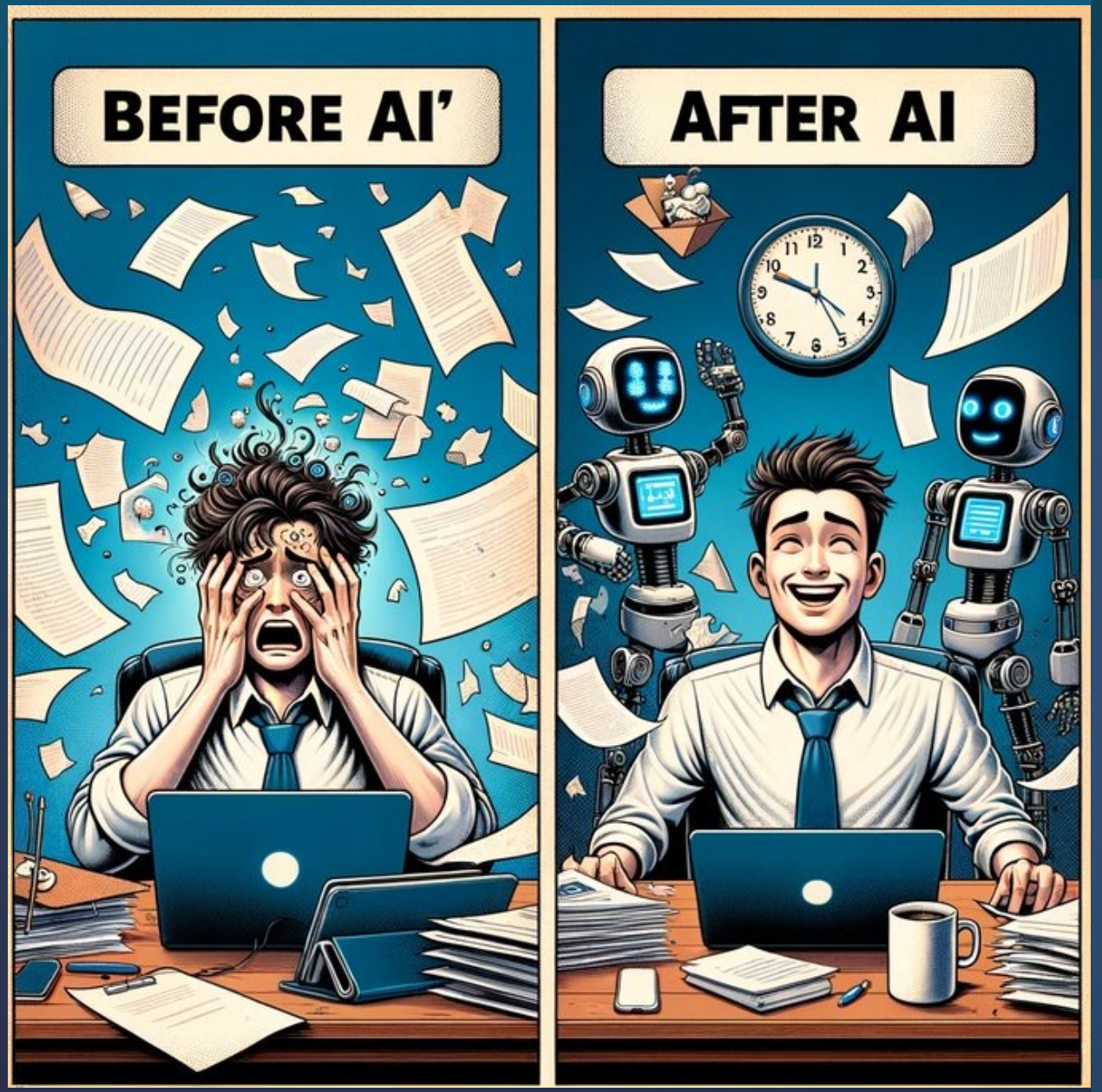
Introduction

Industry talk

What is RAG?

Sneak peek: Agents

Q & A



@Rixhabh__ on X

Why now?

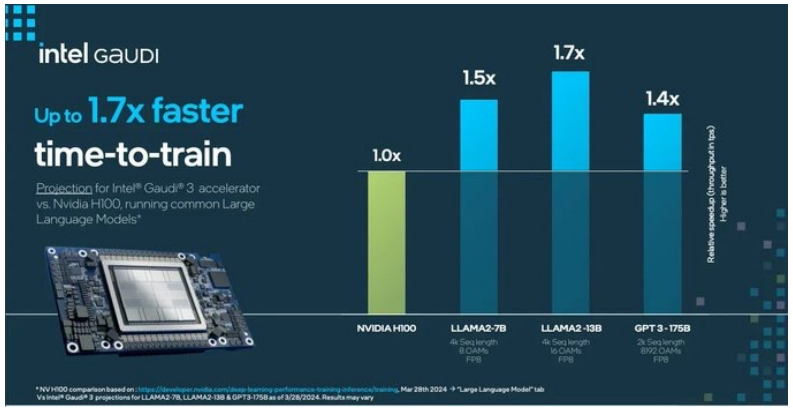
Chief AI Officer
@chiefaioffice

2 hiring trends in AI from Bain & Company:

1. New AI job postings reached a 6-month high in March 2024 at 9800 jobs.

Additionally, new AI jobs are outpacing new software jobs as represented by the pink line

10:48 AM · Apr 9, 2024 · **544** Views



Enterprise AI stats from Zscaler

600%
AI/ML tool usage skyrocketed by nearly 600%, from April 2023 to January 2024

Top 3
The top three blocked AI applications are ChatGPT, OpenAI, and Fraud.net

18.5%
Enterprises block 18.5% of AI and ML transactions, a 577% increase

Apple signed a $50M licensing deal with Shutterstock to acquire AI training data. (@chiefaioffice on X)

Jamie Dimon (CEO and Chairman of JP Morgan Chase) says AI could be as transformative as electricity or the internet (@ CNBC)

Industry exposure (Modern)

Large Language Models (LLMs), Accelerated Computing (Nvidia GPUs), Open Source, Generative Capabilities, Faster software packages, Cuda, and Transformers Architecture (attention mechanism).

Why now?

- Productivity gains through AI (LLMs)
- Copilots (code generation)
- Retrieval Augmented Generation (RAG)
- Copilots for all functions
- Agents is all you need
- Agentic RAG
- Revamp of cloud environments
- Competitive advantage
- Industrial capture
- Accelerated scientific discoveries

AI startups raised

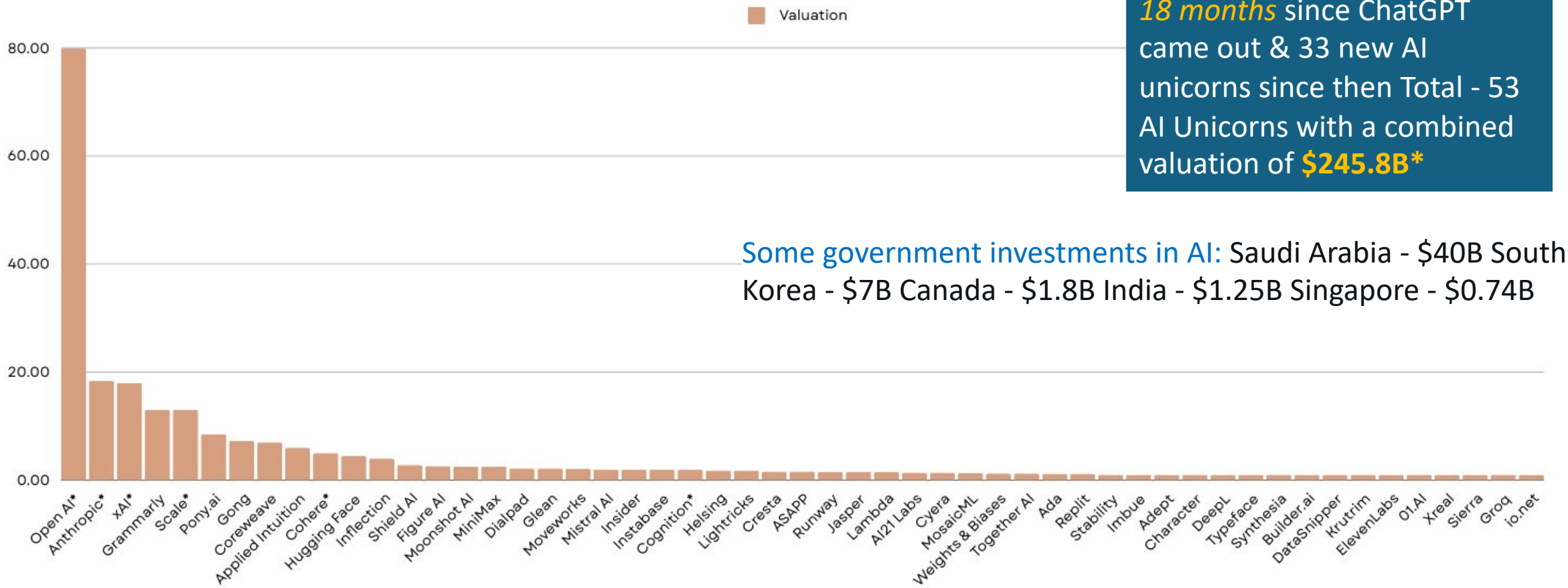
- Feb 2024 : $5.1B
- March 2024 : $6.4B
- Past few days?

New money?

$140m+ (today)
$390m+ (yesterday)
$29m (Apr 17, 2024)
$1.5b (Apr 16, 2024)
$675m (April 10, 2024)
Last week: $1b+

Over $4b+

AI UNICORNS - APRIL 2024



18 months since ChatGPT came out & 33 new AI unicorns since then Total - 53 AI Unicorns with a combined valuation of **$245.8B***

Some government investments in AI: Saudi Arabia - $40B South Korea - $7B Canada - $1.8B India - $1.25B Singapore - $0.74B

@chiefaioffice

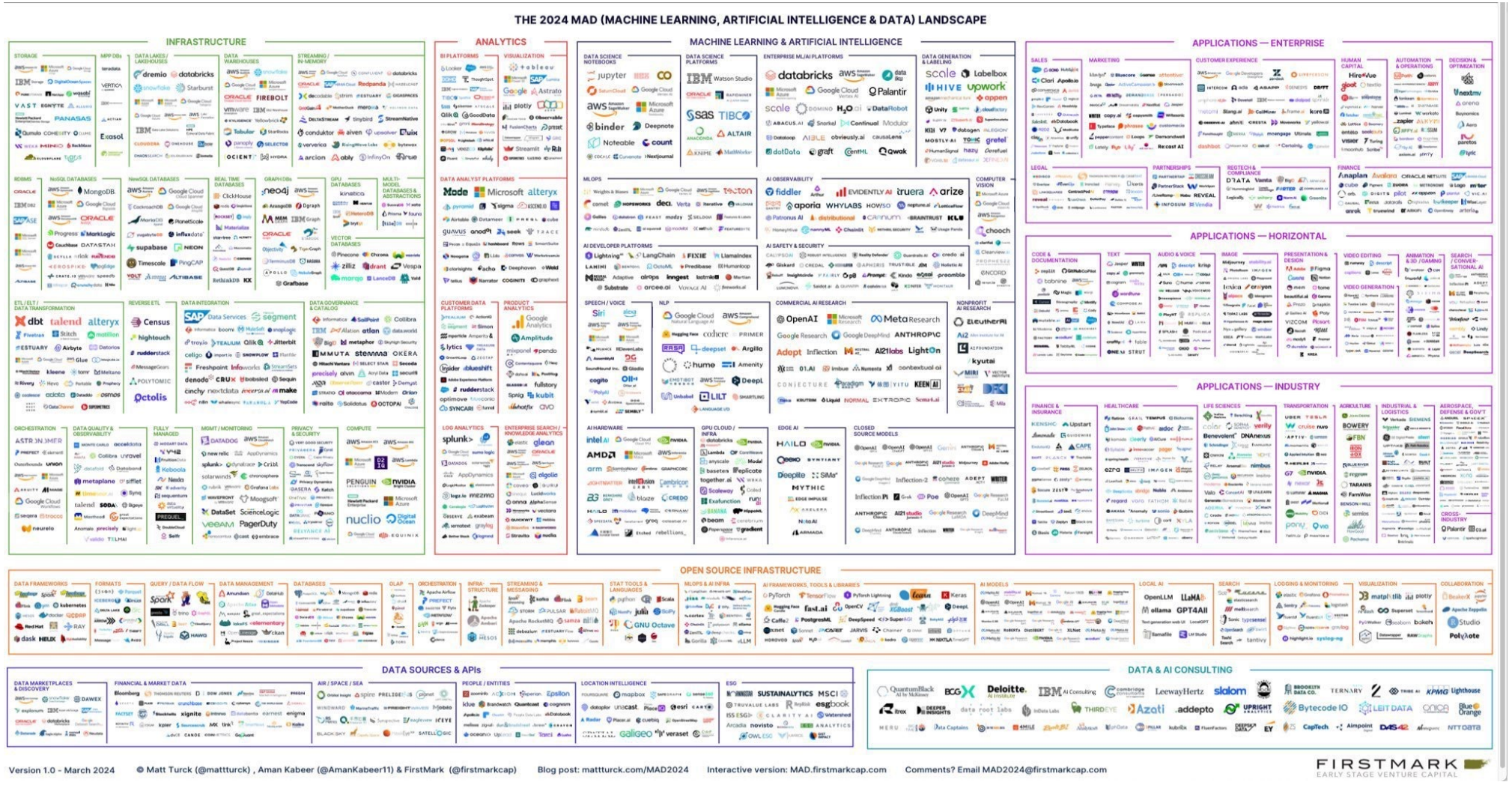
*RUMORED VALUATIONS

Market exposure

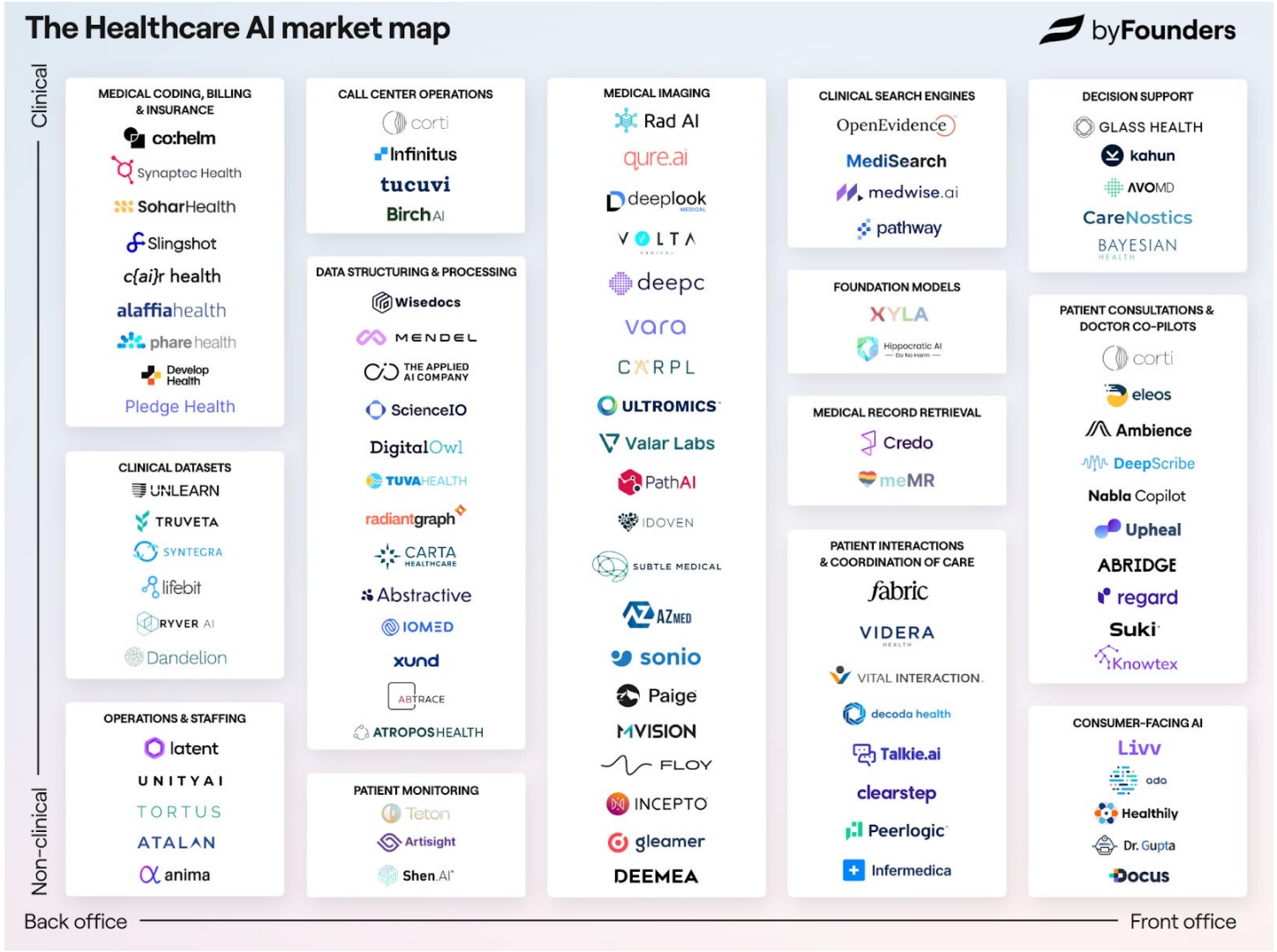
IBM recent report

https://www.linkedin.com/pulse/generative-ai-fueling-sports-entertainment-business-ibm-bkw8e/

Landscape of AI Companies

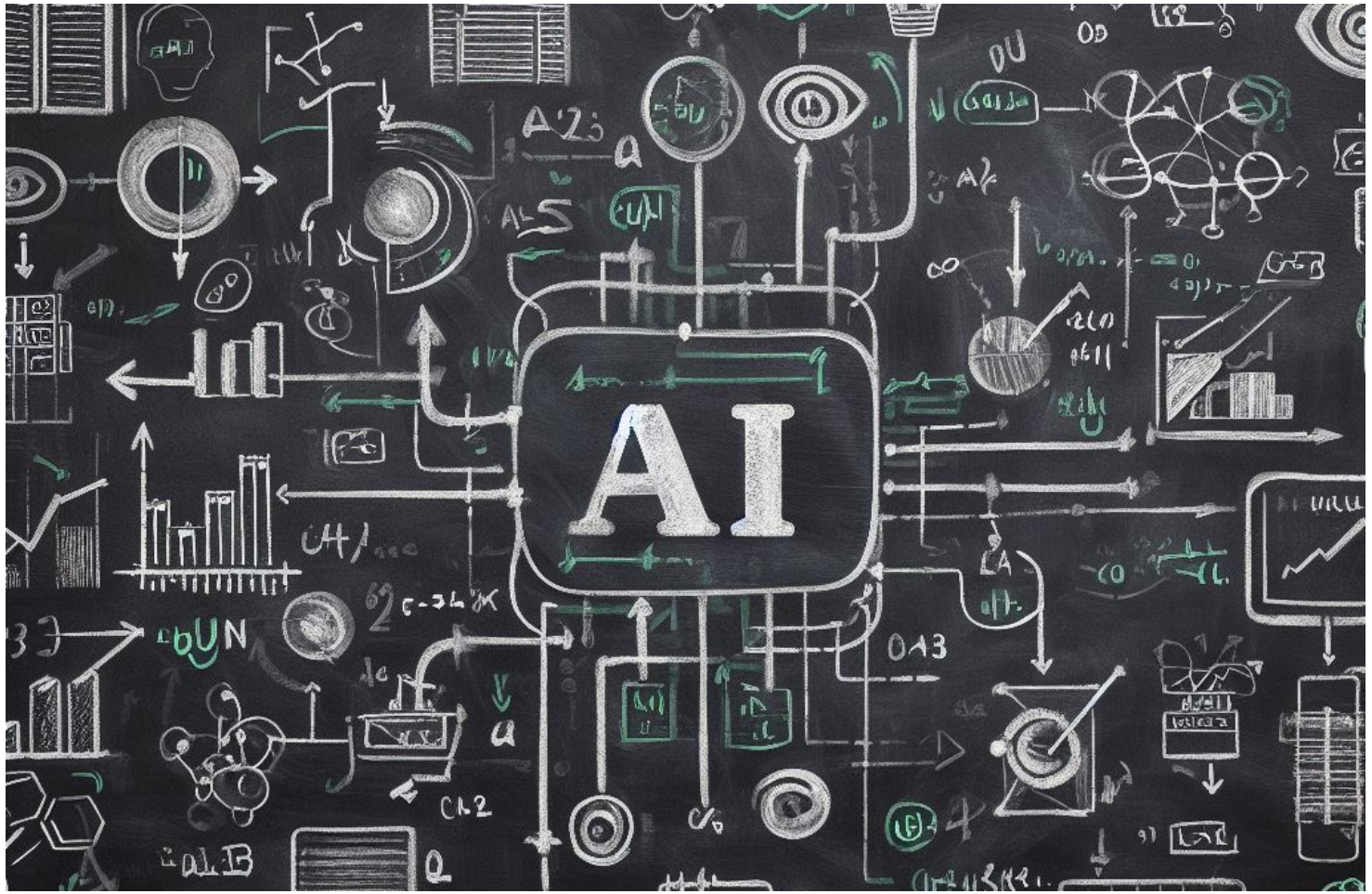


Landscape of AI Companies (Healthcare AI map)



@chiefaioffice on X

What is ML, DL, and AI



AI is not just about tools and prompts. It’s important to have a basic understanding of its key concepts. Today, let's talk about *43 fundamental AI terms*, explained in the most straightforward & easy-to-understand manner.

1. Artificial Intelligence (AI):

Technology that makes machines smart and capable of performing tasks like humans.

2. Machine Learning (ML):

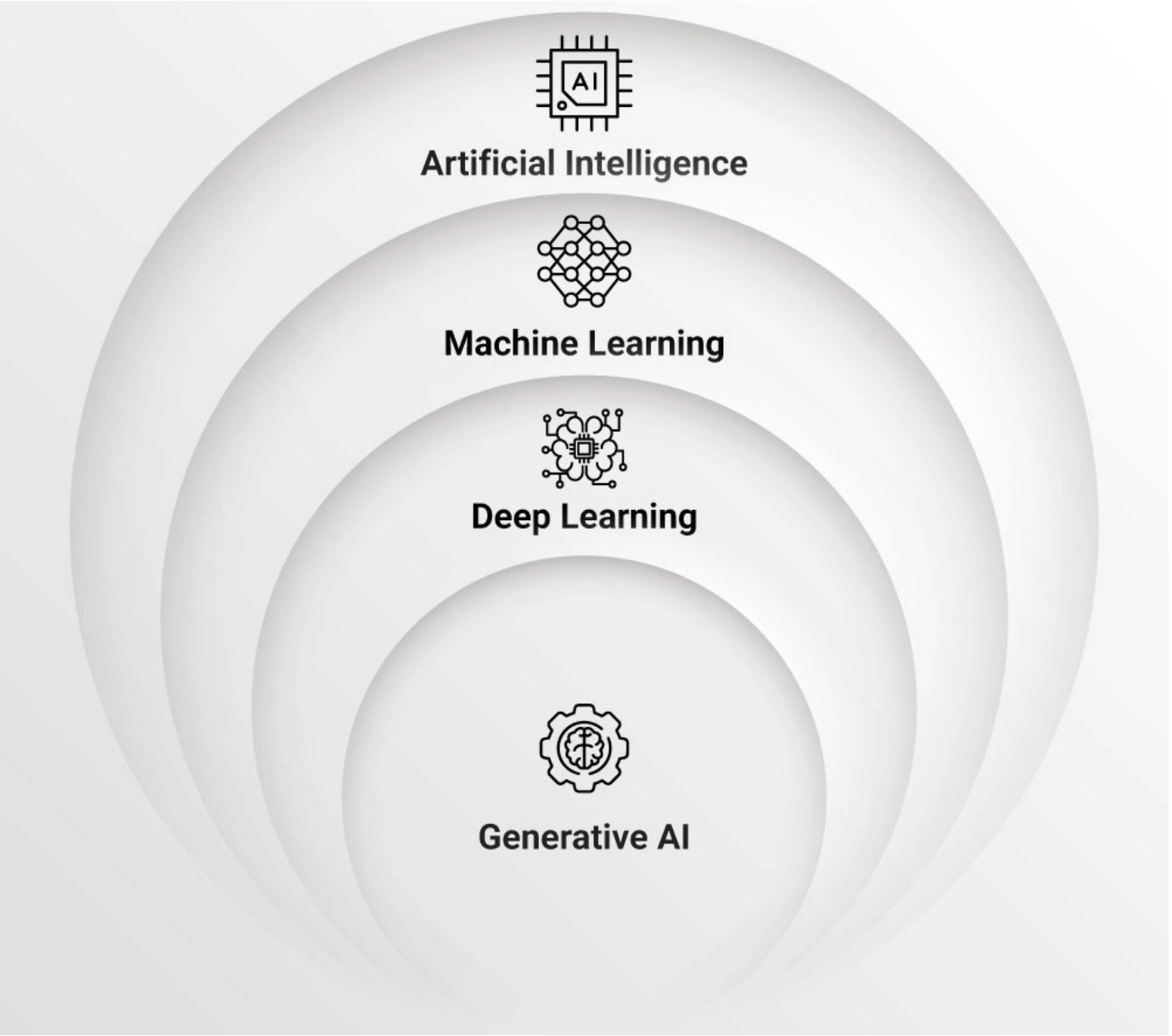
The ability of machines to learn and improve from experience without being *explicitly* programmed.

3. Deep Learning:

A type of machine learning that uses artificial neural networks to imitate how the human brain works.

1. Artificial Intelligence (AI): Technology that makes machines smart and capable of performing tasks like humans. 2. Machine Learning (ML): The ability of machines to learn and improve from experience without being explicitly programmed. 3. Deep Learning: A type of machine learning that uses artificial neural networks to imitate how the human brain works. 4. Natural Language Processing (NLP): The ability of computers to understand and interpret human language. 5. Computer Vision: The ability of computers to understand and interpret visual information from images or videos. 6. Robotics: The field of building and programming robots to perform tasks automatically. 7. Neural Networks: Computer systems inspired by the human brain that learn patterns & make decisions. 8. Big Data: Large amounts of data that can be analyzed by computers to find patterns & insights. 9. Data Science: The study of data to extract meaningful information. 10. Data Mining: The process of discovering patterns & relationships in large datasets. 11. Predictive Analytics: Using historical data to make predictions about future events/outcomes. 12. Supervised Learning: A type of machine learning where models learn from labeled examples. 13. Unsupervised Learning: A type of machine learning where models learn from unlabeled data to find patterns. 14. Reinforcement Learning: Teaching machines to make decisions through trial and error and rewards or punishments. 15. Transfer Learning: Using knowledge from one task to help solve a different but related task. 16. Artificial Neural Networks: Algorithms that mimic the way human brains process info. 17. Convolutional Neural Networks: Neural networks designed to analyze visual data. 18. Recurrent Neural Networks (RNNs): Neural networks that can handle sequential data like language & speech. 19. Generative Adversarial Networks (GANs): 2 neural networks, one generating data and the other evaluating its authenticity. 20. Support Vector Machines (SVMs): Algorithms used for classification & regression tasks. 21. Decision Trees: Models that make decisions by following a series of tree-like structures. 22. Random Forests: Ensembles of decision trees that work together to improve accuracy. 23. K-Nearest Neighbors(KNN): A simple algorithm that classifies data based on similarity to its neighbors 24. Clustering: Grouping data points based on their similarities. 25. Dimensionality Reduction: Simplifying data by reducing number of features while retaining important info. 26. Regression: Predicting a numerical value based on input data. 27. Classification: Assigning categories or labels to data. 28. Overfitting: When a model becomes too specialized in the training data and fails to generalize to new data. 29. Underfitting: When a model is too simple & fails to capture important patterns in the data. 30. Hyperparameters: Settings that control how an ML algorithm learns & makes predictions. 31. Optimization: The process of finding best settings for a machine learning algorithm. 32. Gradient Descent: An optimization algorithm used to update the model's parameters based on the error. 33. Backpropagation: A method for calculating how errors propagate through a neural network. 34. Loss Function: A measure of how well a model is performing. 35. Activation Function: A mathematical function that introduces nonlinearity to neural networks. 36. Batch Normalization: A technique that normalizes input data to speed up training. 37. Dropout: Randomly deactivating some neurons during training to prevent overfitting. 38. Learning Rate: A parameter that controls step size during optimization. 39. Momentum: A technique that helps optimization algorithms converge faster. 40. Early Stopping: Stopping training process early to prevent overfitting when model's performance on validation data declines. 41. Ensemble Learning: Combining predictions of multiple models to improve accuracy. 42. Bias: Systematic errors in a model that cause it to deviate from true values. 43. Variance: Variability of a model's predictions due to sensitivity to fluctuations in training data.

Roots



https://synoptek.com/insights/it-blogs/data-insights/ai-ml-dl-and-generative-ai-face-off-a-comparative-analysis/

Large Language Models (LLMs)

What is a Large Language Model?

A Large Language Model (LLM) is an advanced AI system designed to understand, generate, and interact *using human language*.

The Structure of an LLM

An LLM consists of two core components: a parameters file and a run file. The parameters file contains the neural network's weights, and the run file is the code that activates these parameters. For instance, the Llama-2 70b's parameters file is a massive 140 gigabytes, showcasing the model's complexity.

The Core Function of LLMs

Next Word Prediction

At its heart, an LLM predicts the next word in a sequence. This task, though seemingly simple, necessitates a deep understanding of language and context, enabling the model to generate coherent and relevant text.

https://medium.com/@vidrihmarko/what-is-large-language-model-llm-and-how-does-ai-work-simplified-47c84d6046a7

Attention mechanism in AI

The goal is to enable the AI model to selectively focus on different parts of the input data when producing output.

The attention mechanism generates scores (often using a function of the inputs), determining ***how much focus to place on each data part***. These scores are used to create a **weighted sum of the inputs**, which feeds into the next network layer.

This allows the model to capture ***context and relationships*** within the data that might be missed with traditional, fixed approaches to processing sequences.

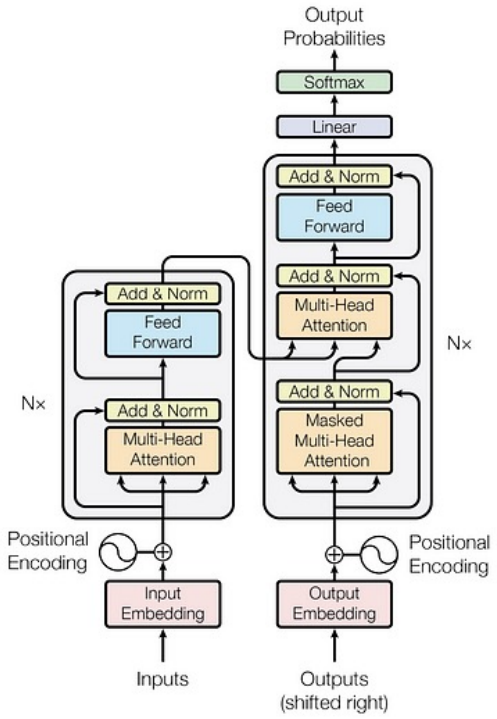
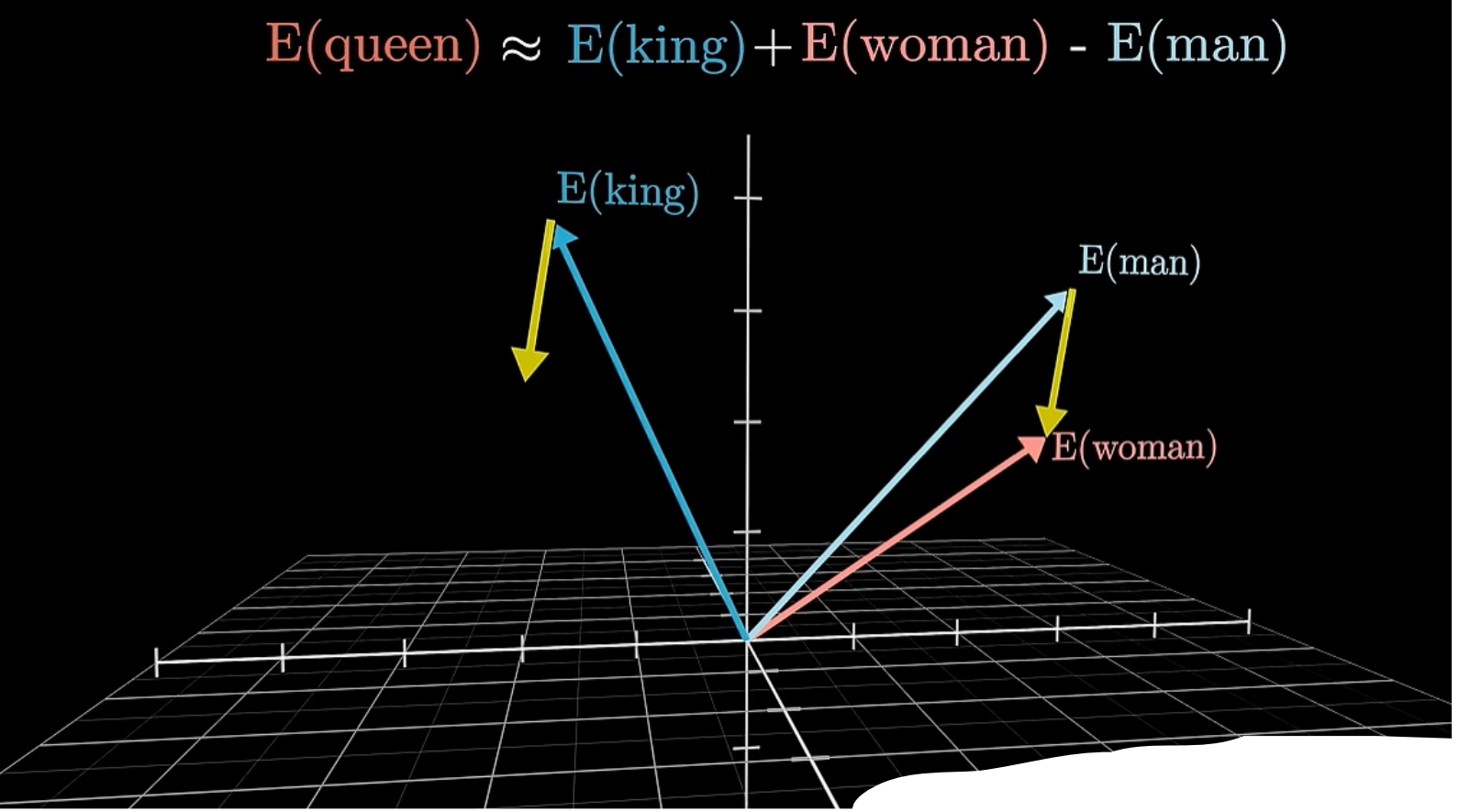
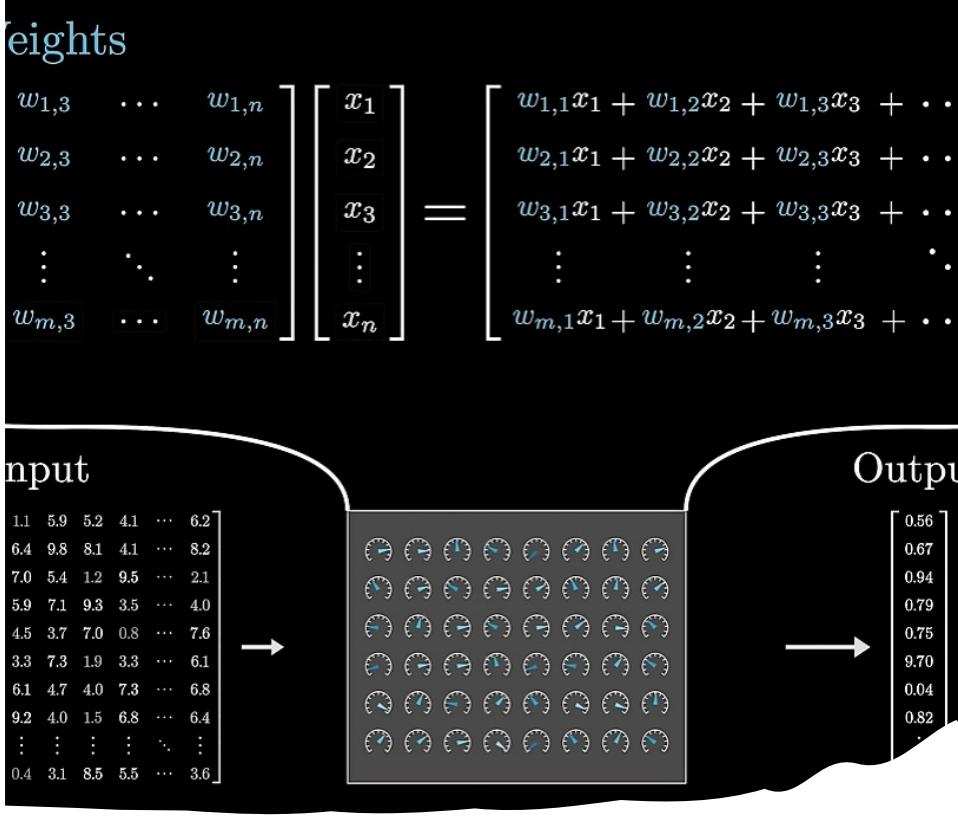
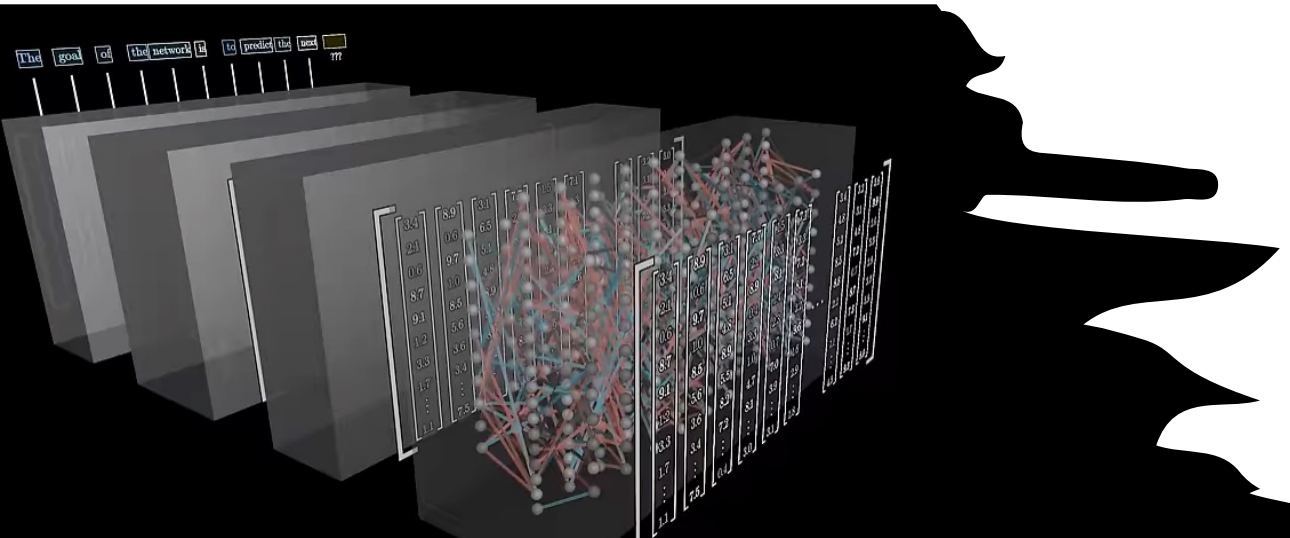


Figure 1: The Transformer - model architecture.

https://medium.com/the-research-nest/explained-attention-mechanism-in-ai-e9bb6f0b0b4d
https://arxiv.org/abs/1706.03762







Intuition

3blue1brown

What is Retrieval Augmented Generation (RAG)?

RAG is a framework for improving model performance by augmenting prompts with *relevant data outside the foundational model*, grounding LLM responses on real, trustworthy information.

https://medium.com/enterprise-rag/an-introduction-to-rag-and-simple-complex-rag-9c3aa9bd017b

Embeddings

The objective of embeddings is to capture ***semantic and syntactic relationships between words***. This helps machines understand and reason about language more effectively.

In other words, the objective was to return **the nearest neighbors** as measured by a similarity metric, which could be:

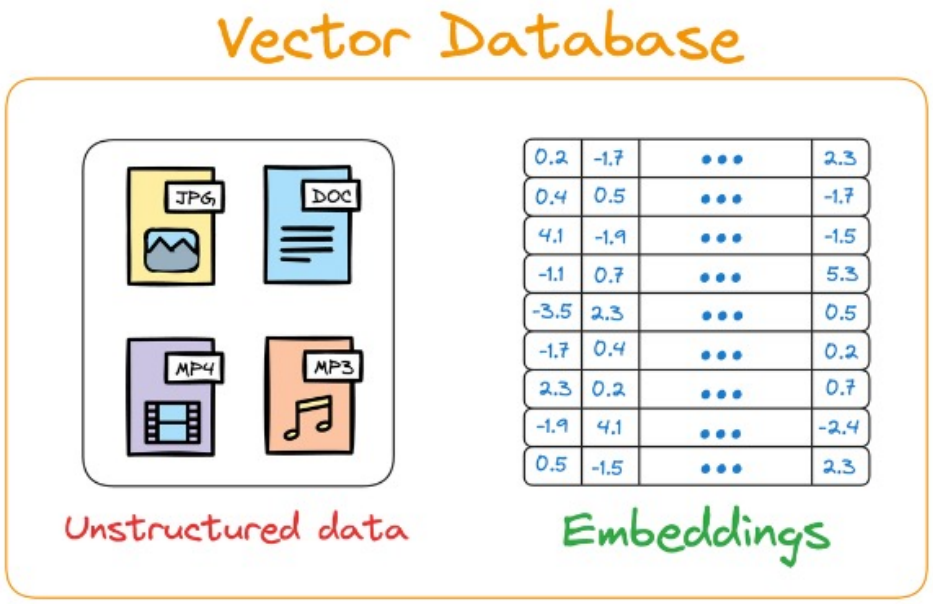
Euclidean distance (the lower the metric, the more the similarity).

Manhattan distance (the lower the metric, the more the similarity).

Cosine similarity (the more the metric, the more the similarity).

https://www.dailydoseofds.com/a-beginner-friendly-and-comprehensive-deep-dive-on-vector-databases/

Embeddings and Vector Database



Instead of relying solely on keywords and following a brute-force search, we can first represent text data in a ***high-dimensional vector space*** and store them in a vector database.

A simple inquiry like:
"What's the weather like today?"

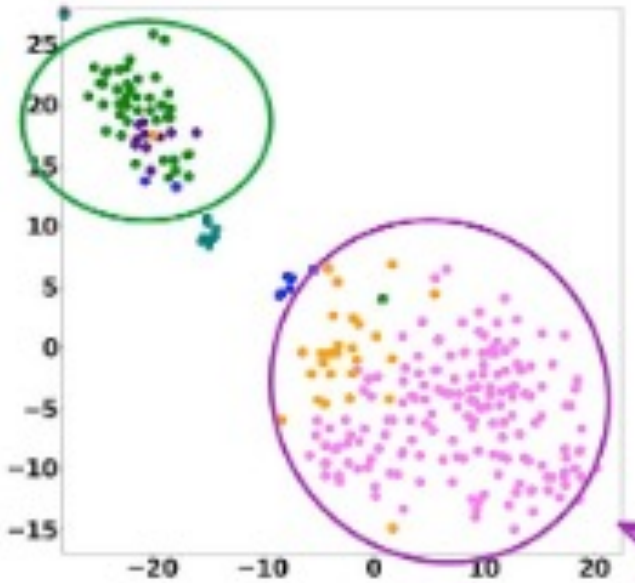
Can be phrased in numerous ways, such as:
"How's the weather today?",
"Is it sunny outside?",
"What are the current weather conditions?".

This linguistic diversity makes traditional keyword-based search methods inadequate.

https://www.dailydoseofds.com/a-beginner-friendly-and-comprehensive-deep-dive-on-vector-databases/

Illustration of vector representation

The word 'bank' when used in water sense

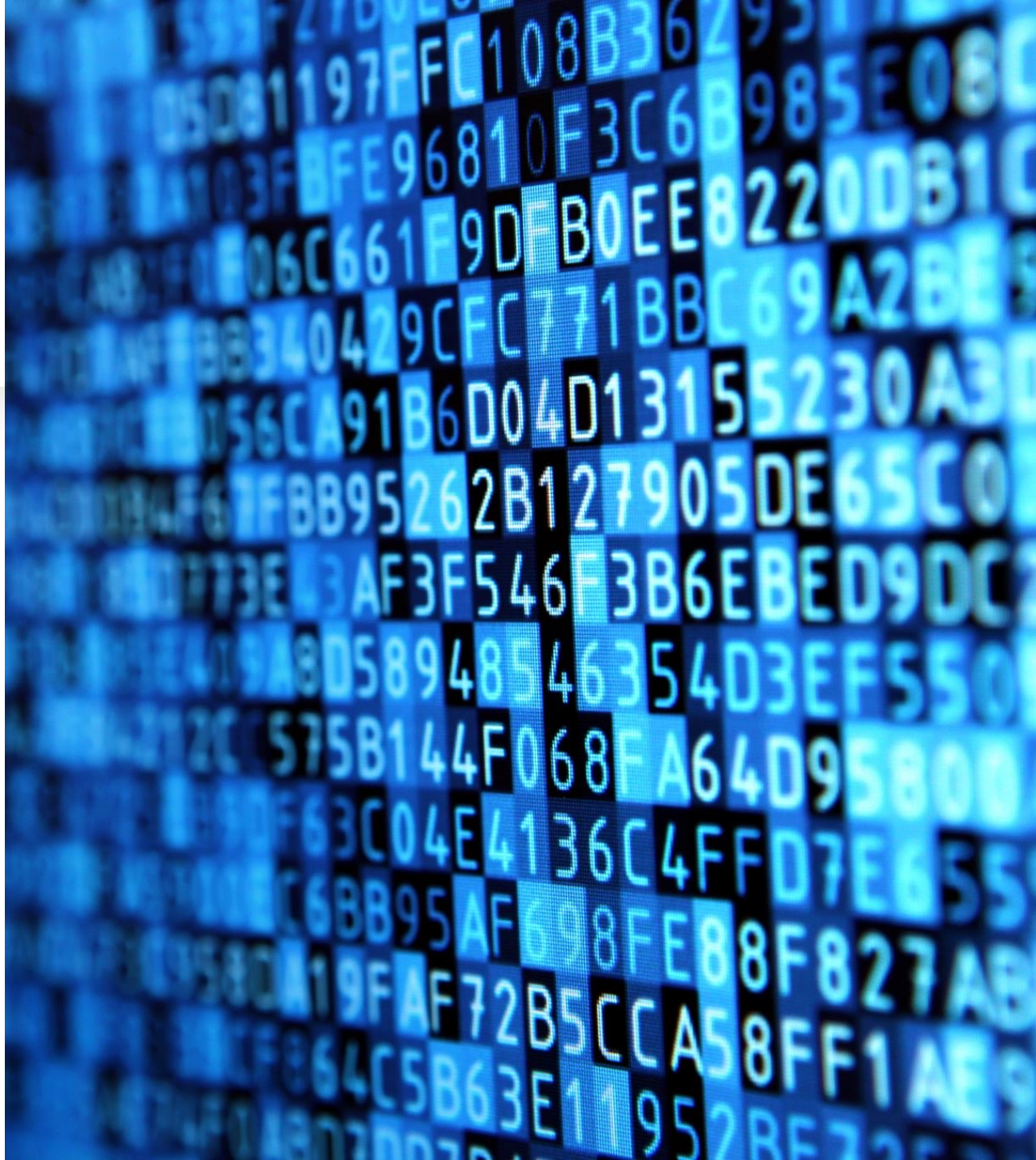


The word 'bank' when used in financial sense

https://www.dailydoseofds.com/a-beginner-friendly-and-comprehensive-deep-dive-on-vector-databases/

Retrieval

When users pose queries, the vector database can compare the vector representation of the query with that of the text data, **even if they don't share the exact same wording.**



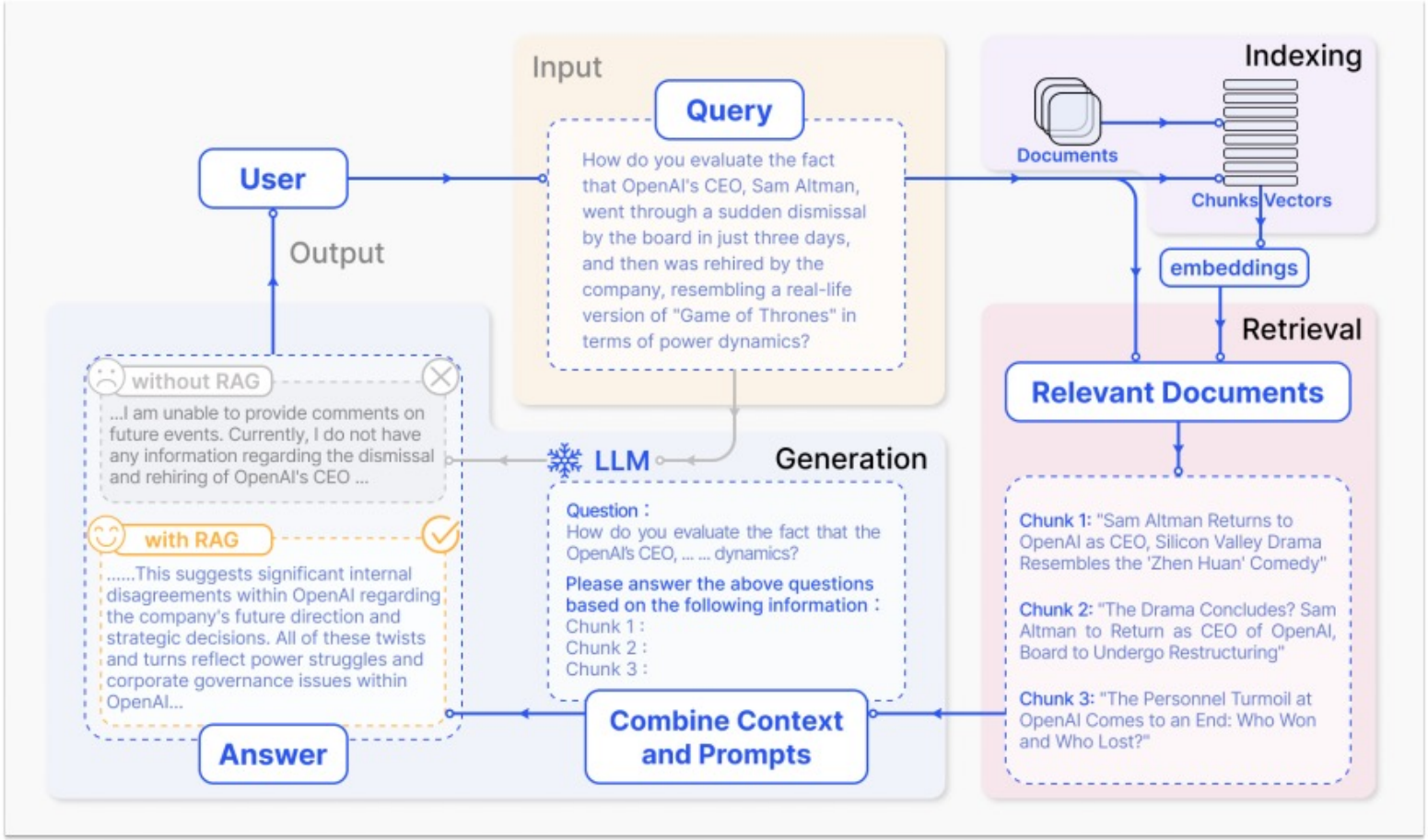
Contextual view – Typical RAG

It mainly consists of 3 steps:

1) Indexing : Documents are split into chunks, encoded into vectors, and stored in a vector database

2) Retrieval : Retrieve the Top k chunks most relevant to the question based on semantic similarity

3) Generation : Input the original question and the retrieved chunks together into LLM to generate the final answer.



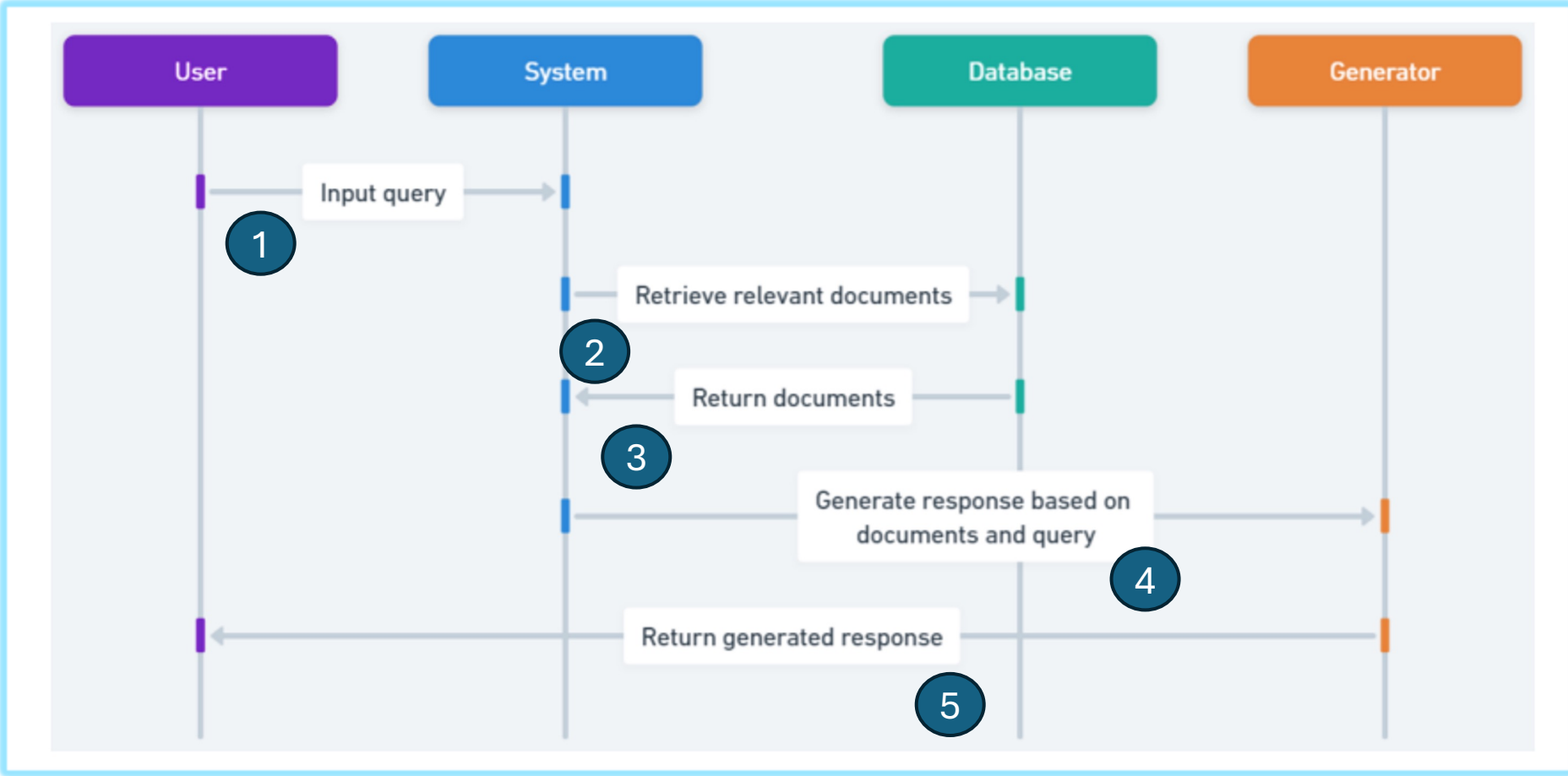
https://arxiv.org/pdf/2312.10997.pdf

System view - Typical RAG

User question -> Automatic search in a database for relevant information ->

Give back the question +

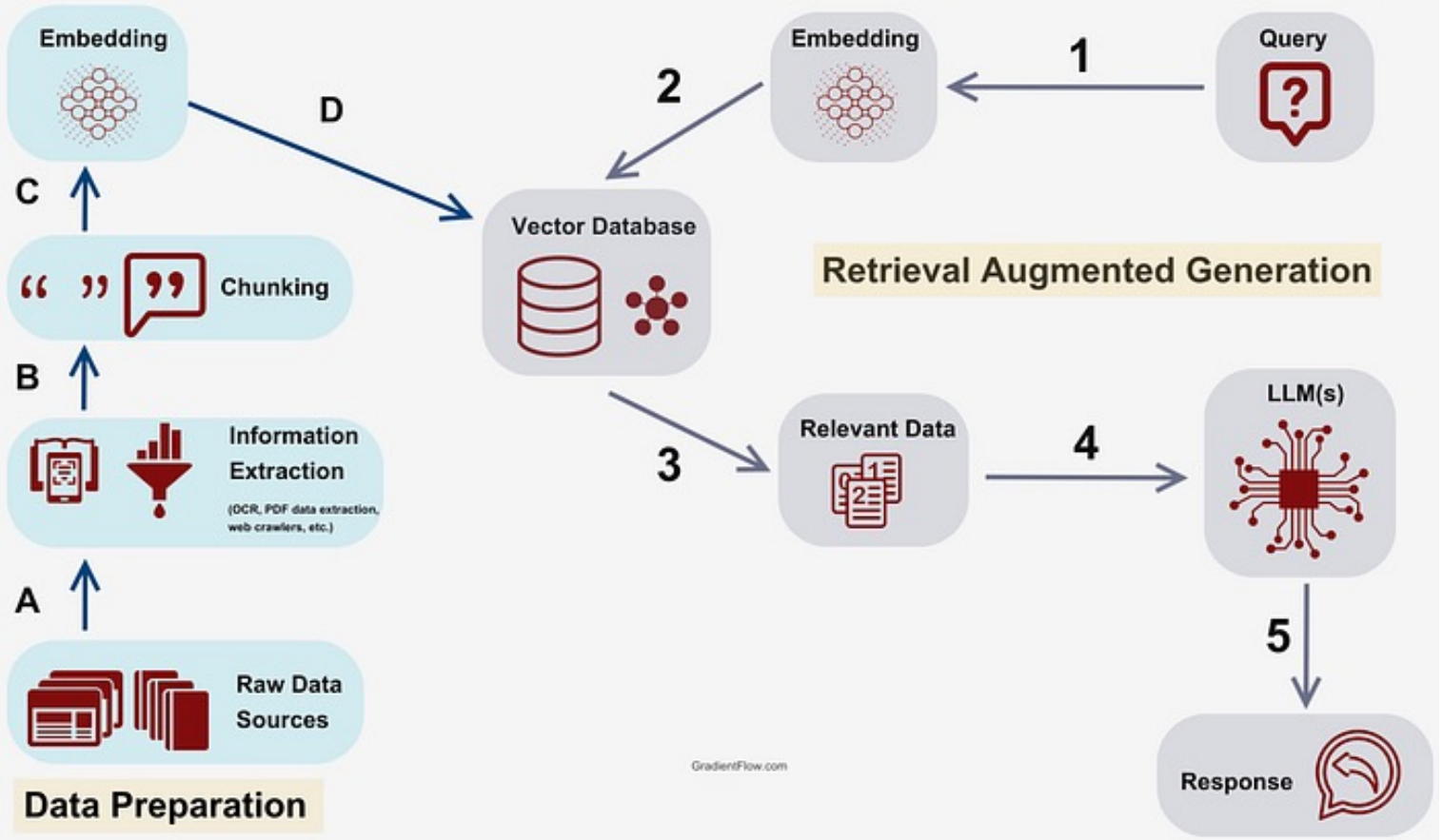
Relevant info found from dataset to LLM -> Answer user



https://arxiv.org/pdf/2404.01037.pdf

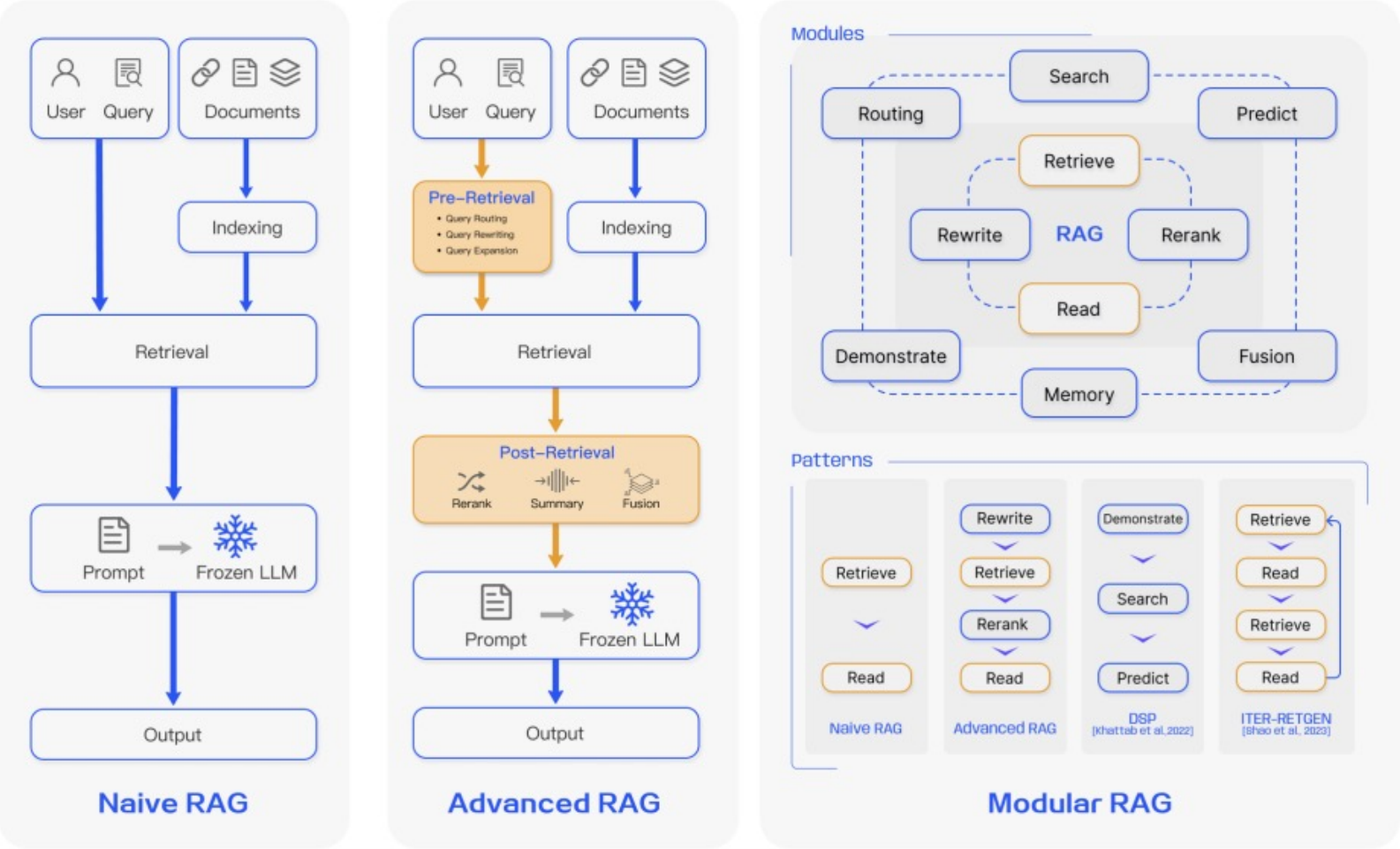
Implementation view - Typical RAG

- Data quality
- Data preparation
- Data sources
- Metadata
- Additional Context and Knowledge



https://gradientflow.com/techniques-challenges-and-future-of-augmented-language-models/

Naive, Advanced, and Modular RAG

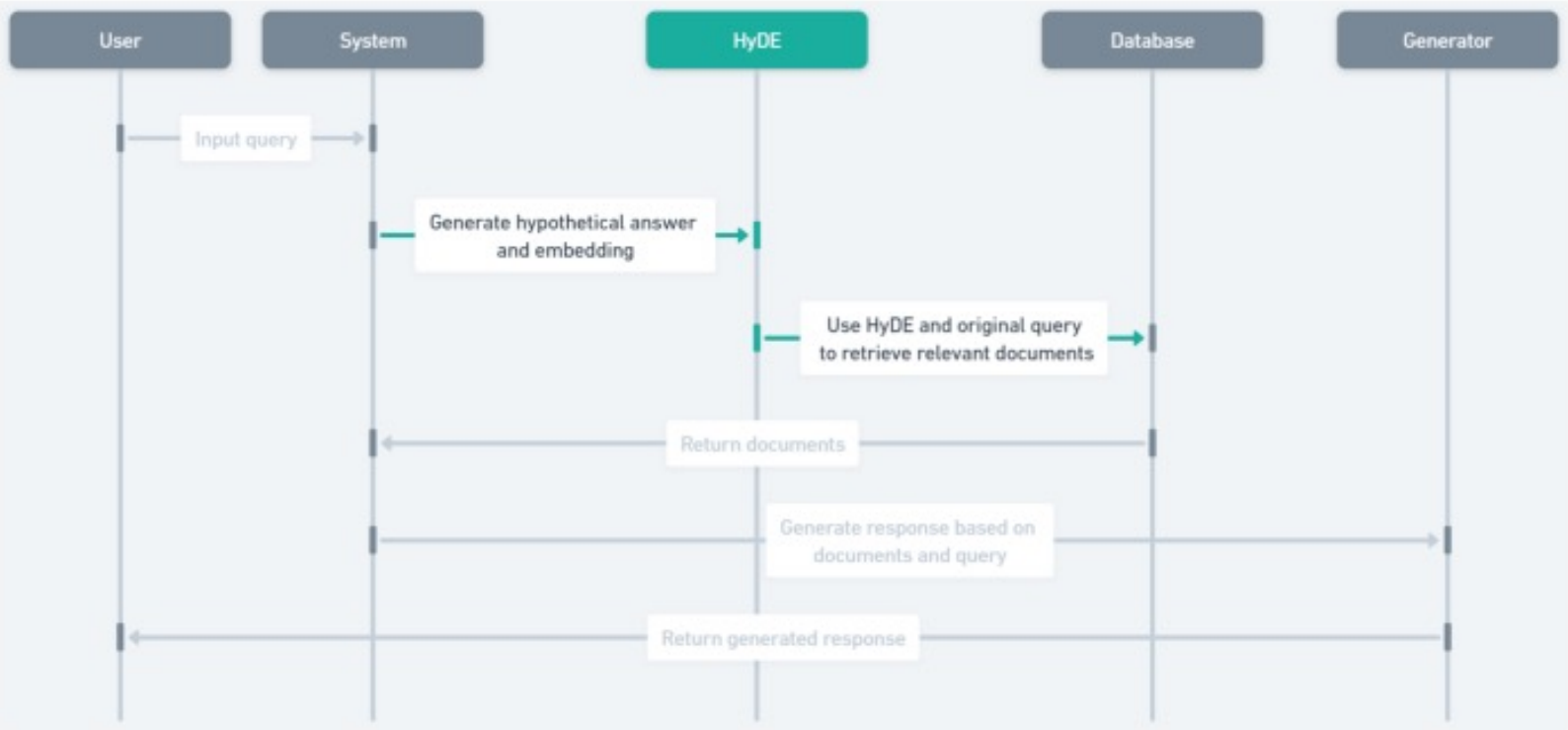


Multiple optimization strategies around pre-retrieval and post-retrieval.

It is iterative and adaptive retrieval.

https://arxiv.org/pdf/2312.10997.pdf

Hypothetical Document Embedding HyDE



The Hypothetical Document Embedding (Gao et al., 2022) technique enhances the document retrieval by leveraging LLMs to generate a hypothetical answer to a query.

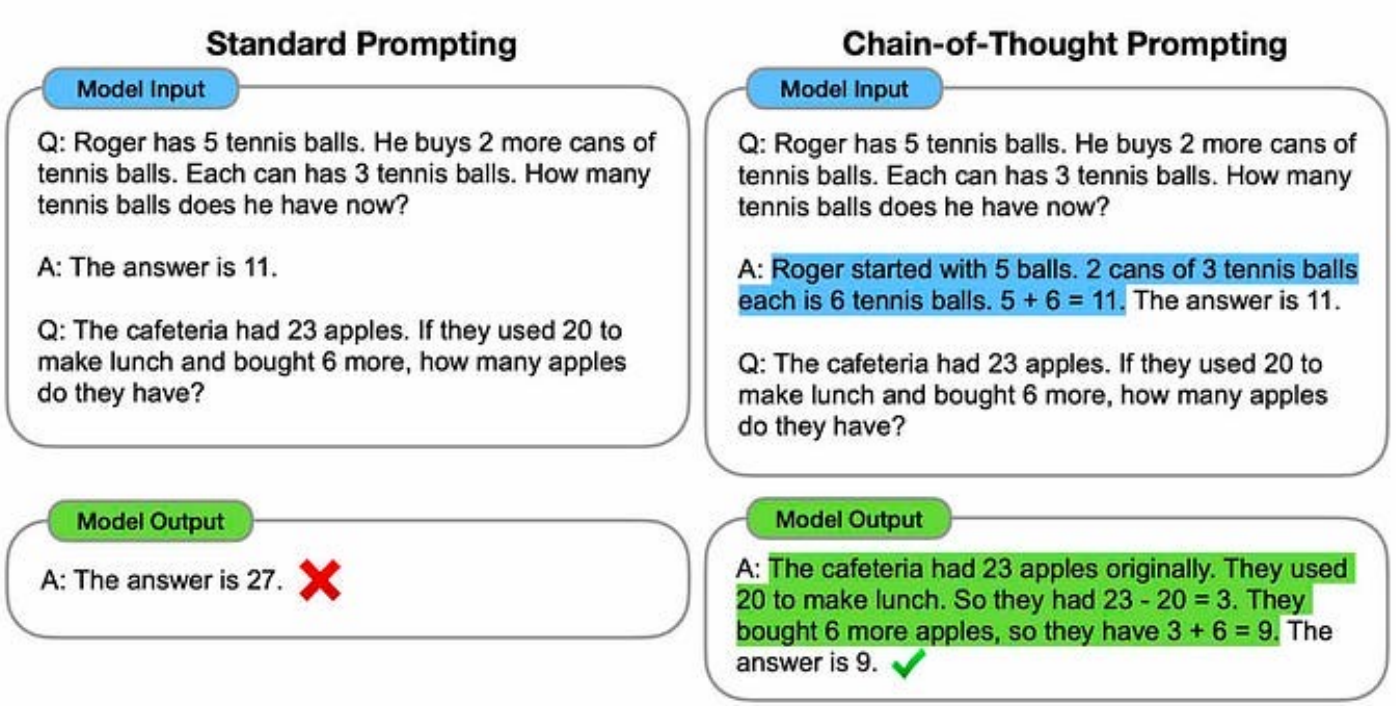
HyDE capitalizes on the ability of LLMs to produce **context-rich answers**, which, once embedded, serve as a powerful tool to refine and focus document retrieval efforts.

https://arxiv.org/pdf/2404.01037.pdf

Key implementation considerations

Chain of thought

Chain of thought (CoT) is a common technique where you break down a complex problem into smaller, more manageable parts and then solve these parts step by step. CoT helps evolve the reasoning in Large Language Models.



Explicit Reasoning: CoT prompts encourage the model to "think aloud," detailing each step of its thought process as it works towards a solution.

Improved Comprehension: By breaking down a problem into intermediate steps, the model can tackle more complex questions than it could by attempting to jump directly to an answer.

Enhanced Accuracy: This step-by-step approach can lead to more accurate outcomes, as it reduces the model's tendency to "hallucinate" or make unsupported leaps in logic.

https://medium.com/@luxananda/6-llm-techniques-of-prompt-engineering-chain-of-thoughts-cot-part-1-a10a76a474e6

RAG – implementation considerations

Wrappers, flow, and model serving

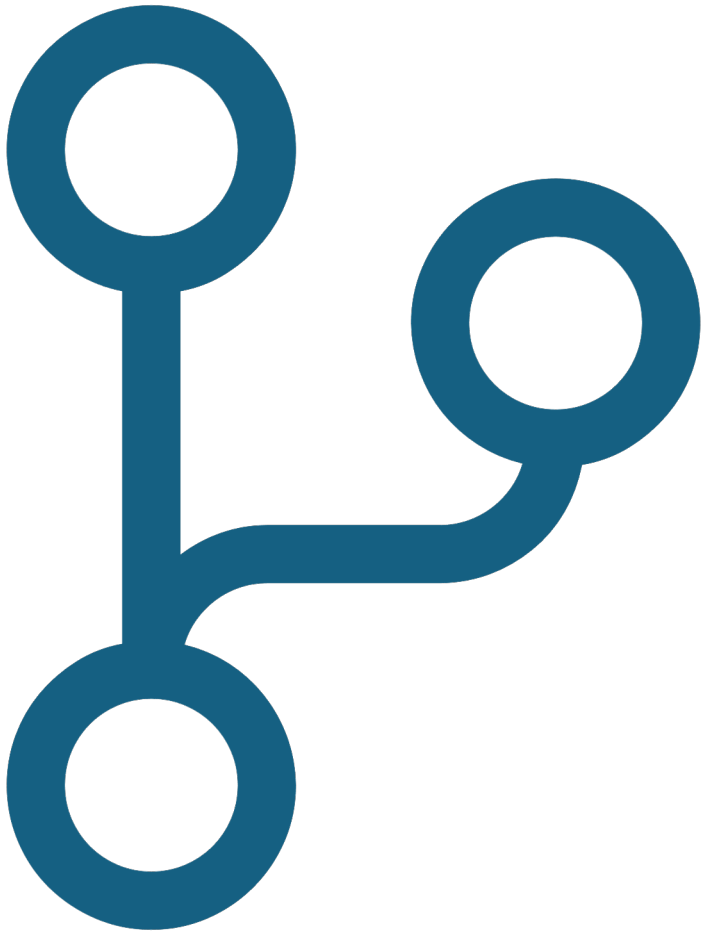
- Langchain
- Lllamaindex
- Ollama
- Huggingface
- OpenAI
- Others

Key packages

- Ragas
- Instructor
- Pydantic
- PyTorch
- PyTorch Lightening
- Skorch
- Function calling (Important)
- Others

RAG - challenges

- Information loss
- World knowledge
- Multi-hop Q & A
- Semantic Search
- Hallucinations
- Bias
- Complexity for scale



Jerry Liu, CEO of LlamaIndex on RAG:

If you think about it, rag is basically prompt engineering, because you're basically figuring out a way to put context into the prompt.

It's just a programmatic way of prompt engineering.

It's a way of prompting so that you actually get back some context [from a database of yours, and in an automated way].

https://pub.towardsai.net/what-is-rag-f790bde23d6f

Retrieval Precision

Hypothetical Document Embedding (HyDE) and LLM reranking significantly enhance retrieval precision.

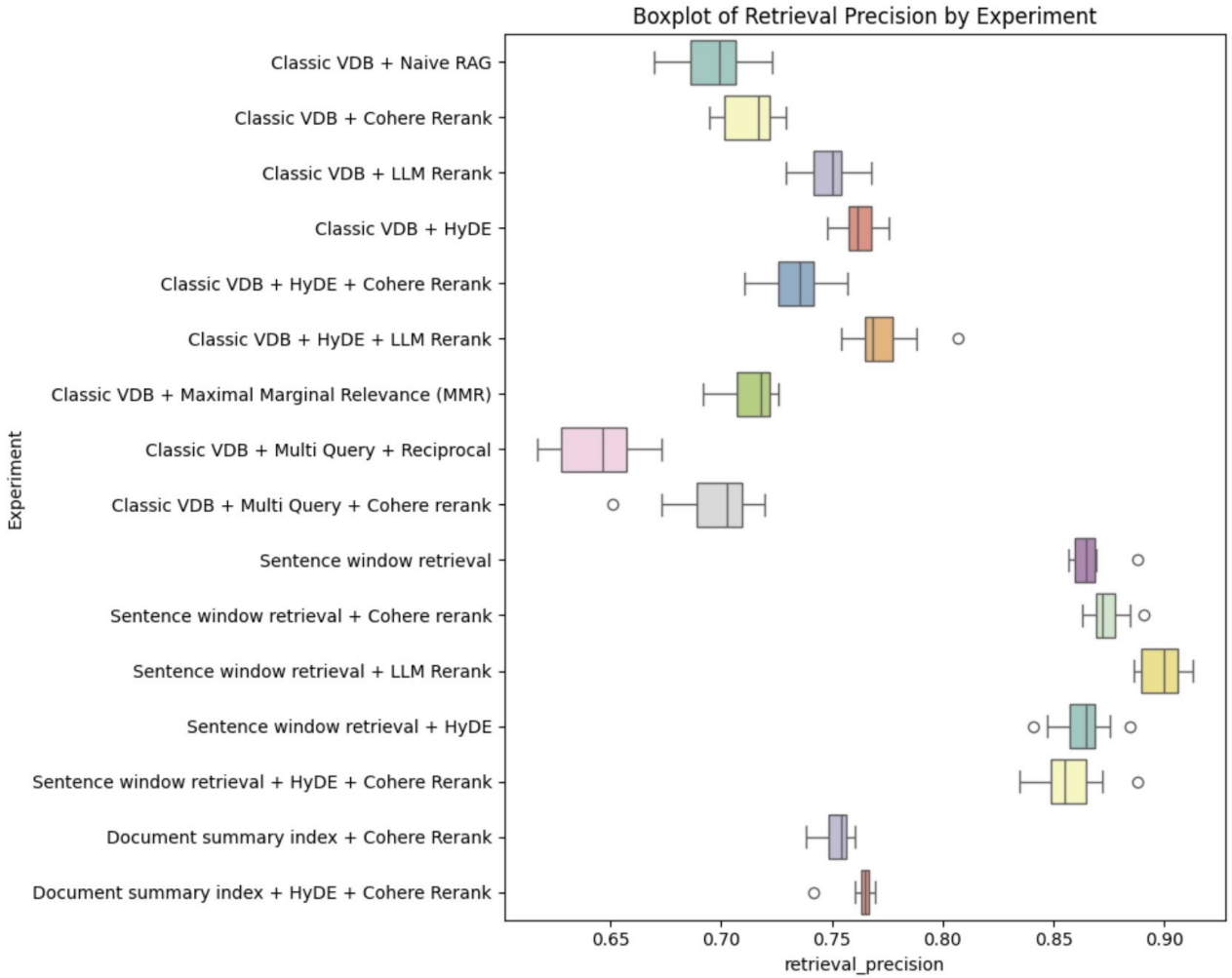
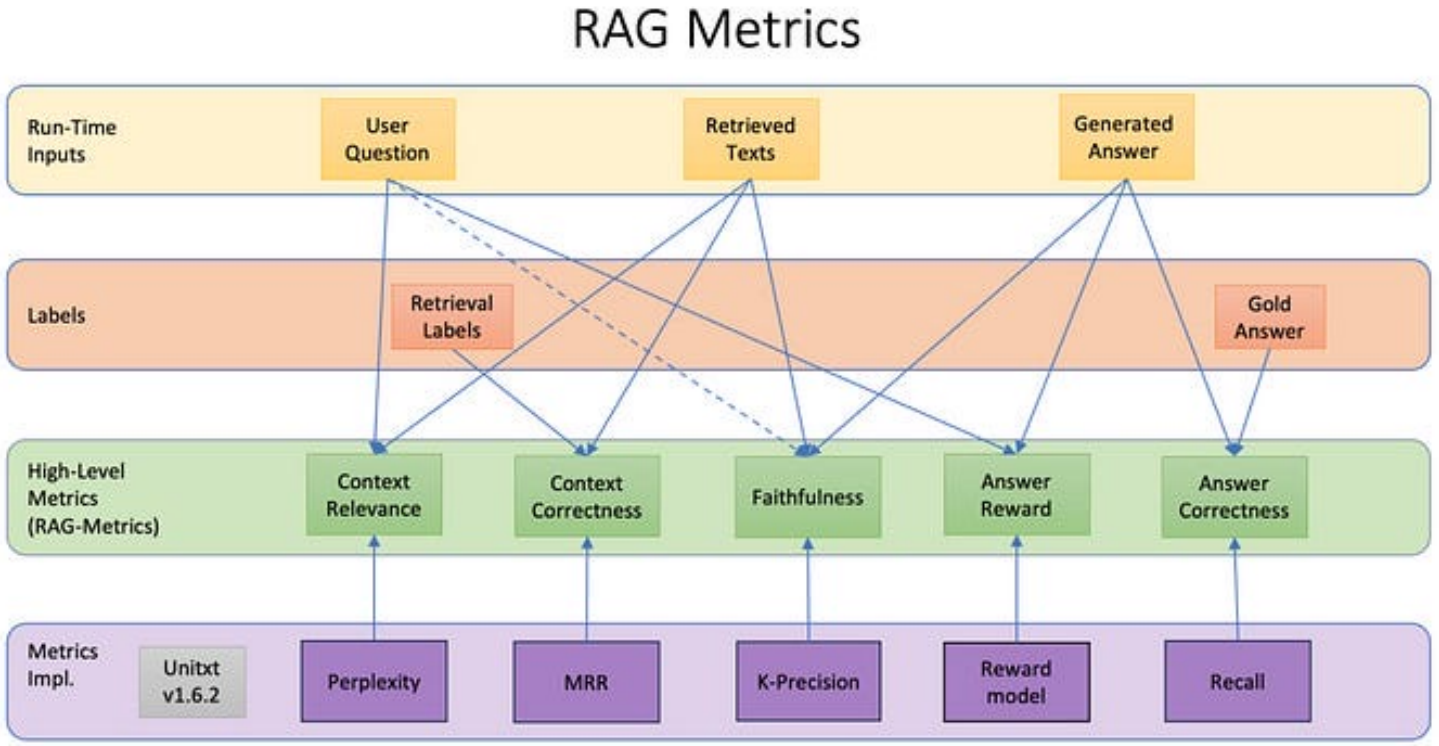


Figure 7: Boxplot of Retrieval Precision by Experiment. Each boxplot demonstrates the range and distribution of retrieval precision scores across different RAG techniques. Higher median values and tighter interquartile ranges suggest better performance and consistency.

https://arxiv.org/pdf/2404.01037.pdf

RAG Evaluation Metrics



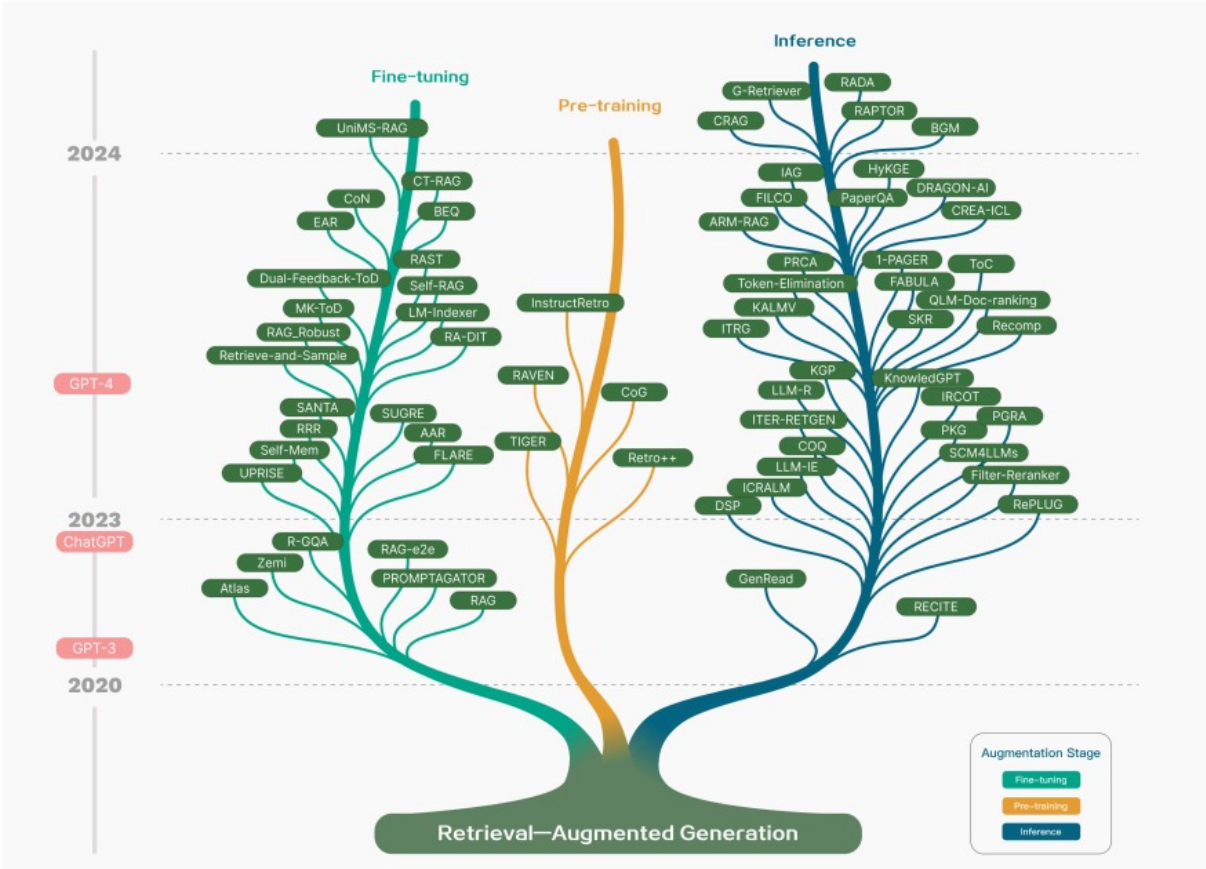
Context Relevance: This is a reference-less metric gauging the **relevance of the retrieved texts** to answering the user question. The metric range is [0, 1], where higher is better.

Context Correctness: This is a reference-based metric **reflecting the rank of the ground-truth text in the retrieved texts**. The metric range is [0, 1] where higher is better.

Faithfulness: This is a reference-less metric gauging **the groundedness of the generated answer in the retrieved texts**. The metric range is [0, 1], where higher is better.

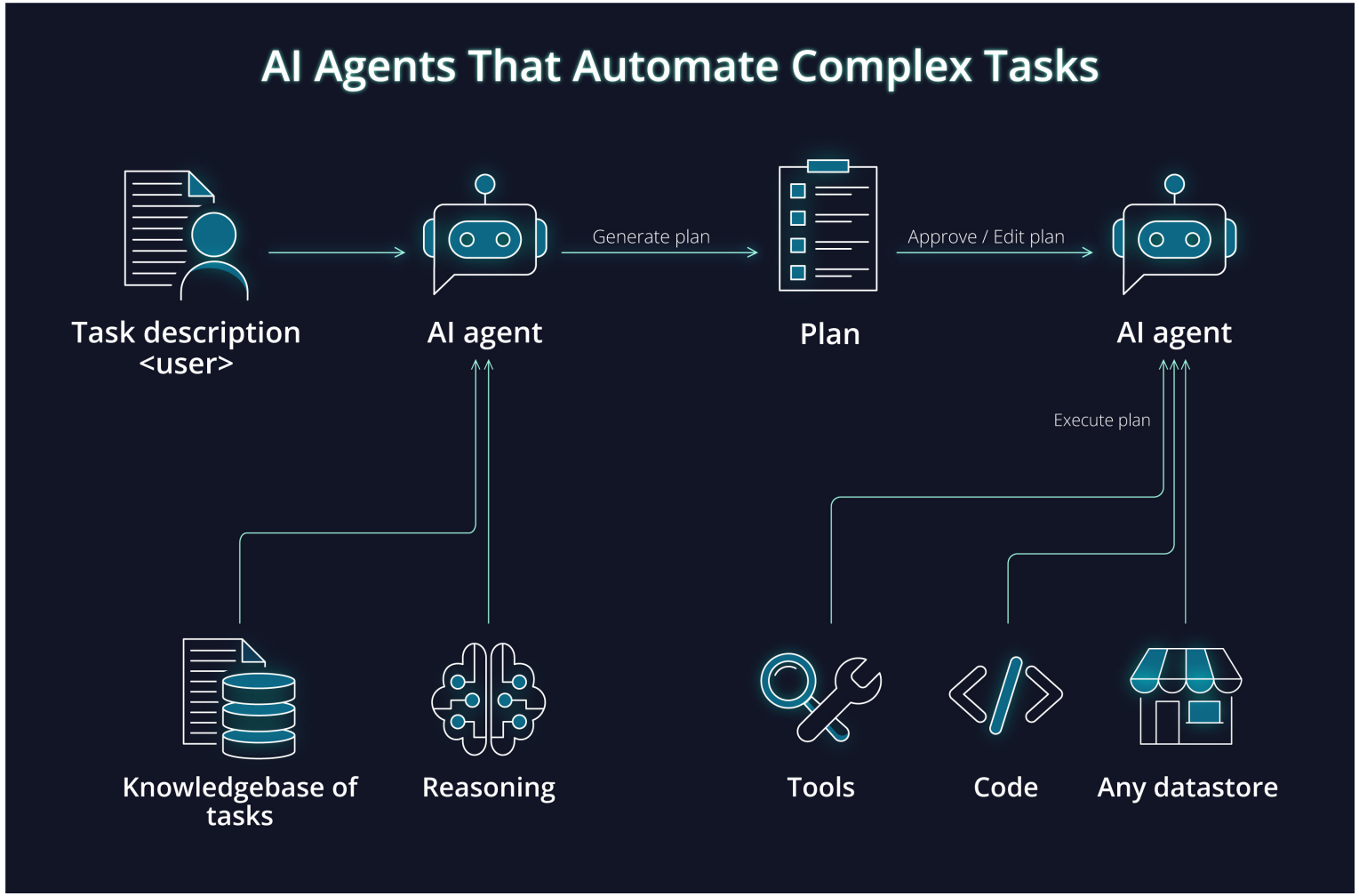
https://medium.com/@ugurozker/rag-implementation-with-llama2-13b-chat-llm-model-742f3663b07b

Technology Tree of RAG research

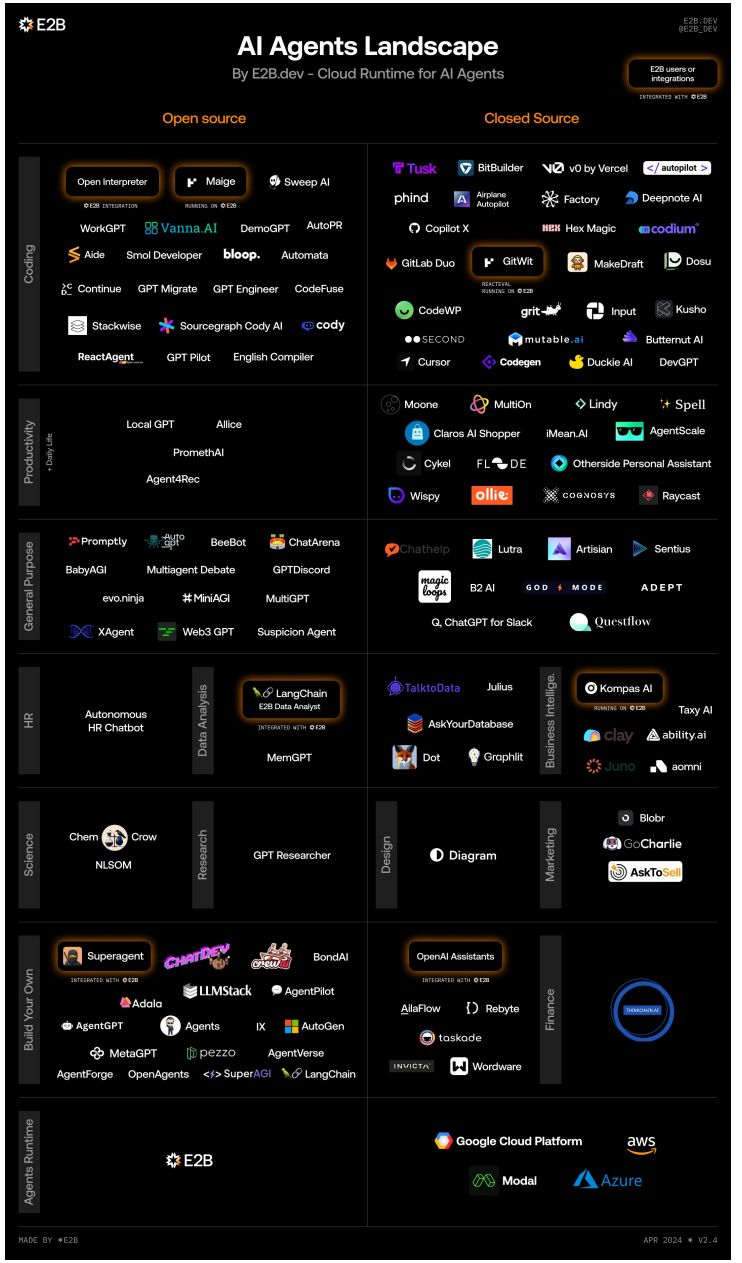


https://arxiv.org/pdf/2312.10997.pdf

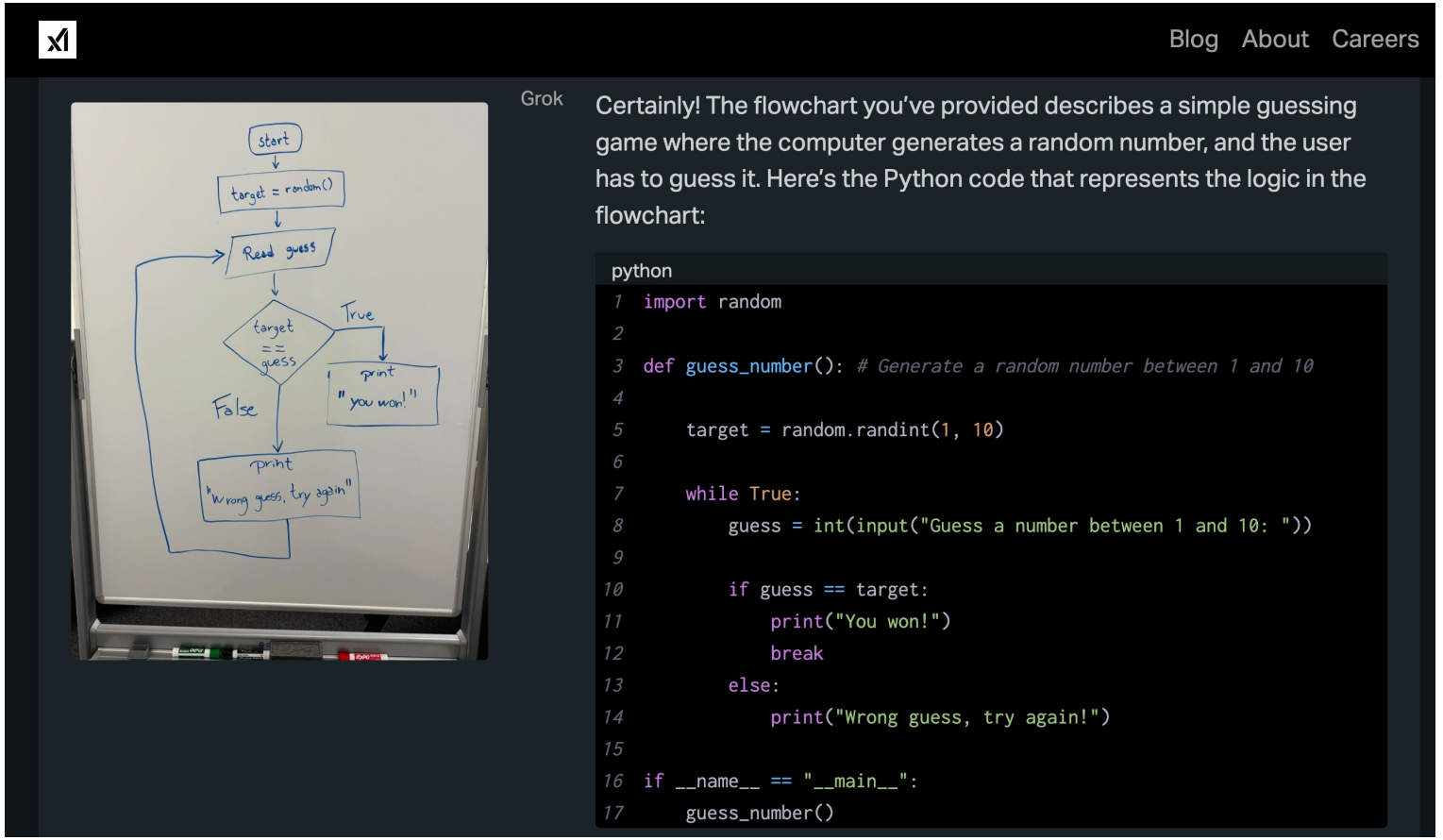
AI Agents are everywhere



Landscape of AI Agents



Code via flowchart (example)



xAI grok 1.5

My favorite : Devin

Cognition
@cognition_labs

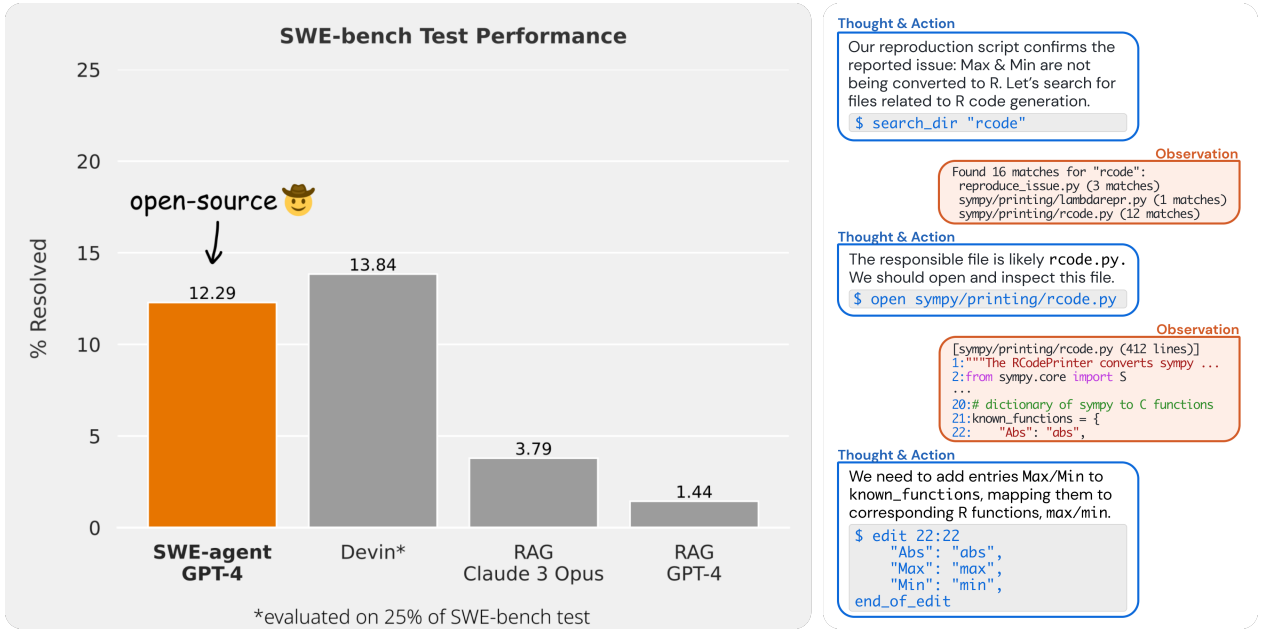
Today we're excited to introduce **Devin, the first AI software engineer**.

Devin is the new state-of-the-art on the SWE-Bench coding benchmark, has successfully passed practical engineering interviews from leading AI companies, and has even completed real jobs on Upwork.

Devin is an autonomous agent that solves engineering tasks through the use of its own shell, code editor, and web browser.

When evaluated on the **SWE-Bench** benchmark, which asks an AI to resolve GitHub issues found in real-world open-source projects, Devin correctly resolves 13.86% of the issues unassisted, far exceeding the previous state-of-the-art model performance of 1.96% unassisted and 4.80% assisted.

SWE-agent from Princeton University



SWE-agent

John Yang*, Carlos E. Jimenez*, Alexander Wettig
Shunyu Yao, Karthik Narasimhan, Ofir Press
Princeton University

- SWE-agent turns LMs (e.g. GPT-4) into software engineering agents that can fix bugs and issues in real GitHub repositories.
- On SWE-bench, SWE-agent resolves **12.29%** of issues, achieving the state-of-the-art performance on the full test set.
- SWE-agent is built and maintained by researchers from Princeton University.

https://github.com/princeton-nlp/SWE-agent

Q & A