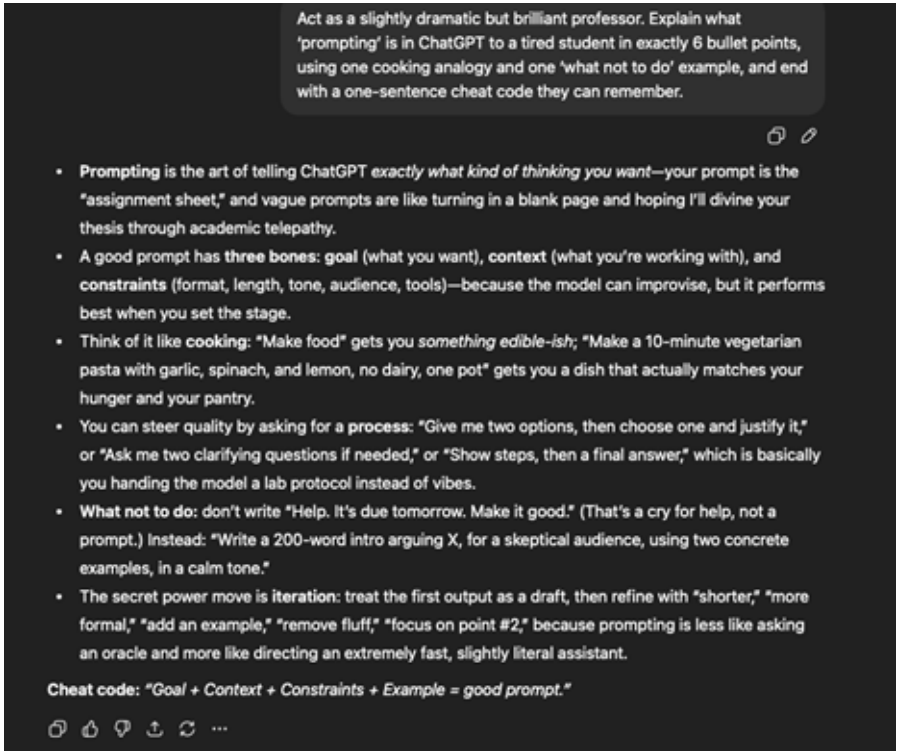
Retrieval-Augmented Generation

Overview

1. Why RAG (LLM limits)
2. End-to-end RAG pipeline + quality levers
3. Retrieval strategies in RAG (hybrid, rerank, query rewrite)
4. Context construction (packing, dedup, citations)
5. Evaluation + debugging playbook
6. Safety (prompt injection, ACLs)
7. Advanced: iterative/token-level retrieval + long-context tradeoffs (optional/appendix)

Start with a simple prompt based interaction



RAG Example with Simple Prompt

Retrieved Document

Knowledge: Arctic Ocean. Although over half of Europe's original forests disappeared through the centuries of deforestation, Europe still has over one quarter of its land area as forest, such as the broadleaf and mixed forests, taiga of Scandinavia and Russia, mixed rainforests of the Caucasus and the Cork oak forests in the western Mediterranean. During recent times, deforestation has been slowed and many trees have been planted. However, in many cases monoculture plantations of conifers have replaced the original mixed natural forest, because these grow quicker. The plantations now cover vast areas of land, but offer poorer habitats for many European
Question: As of 2015, since 1990 forests have _____in Europe and have _____in Africa and the Americas.
A. "increased, increased" B. "increased, decreased" C. "decreased, increased" D. "decreased, decreased"
Answer: B

Complete Prompt

Will cover how this idea applies to general tool-use & reasoning

https://arxiv.org/abs/2301.12652

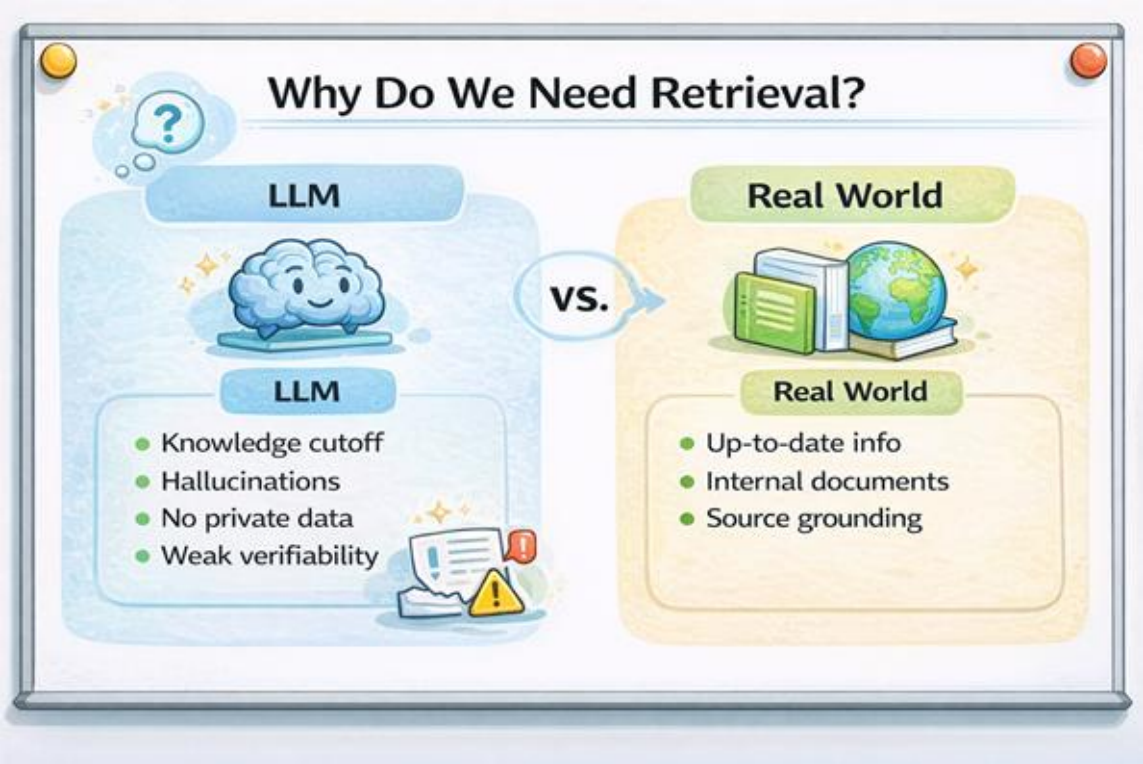
Prompts: interface between humans and LLMs

A prompt usually contains some combination of:

- Instructions (what you want the model to do)
- Context (background information)
- Data (text, code, tables, documents)
- Constraints (format, style, length, rules)

The model does not “know” what you want unless it is expressed in the prompt.

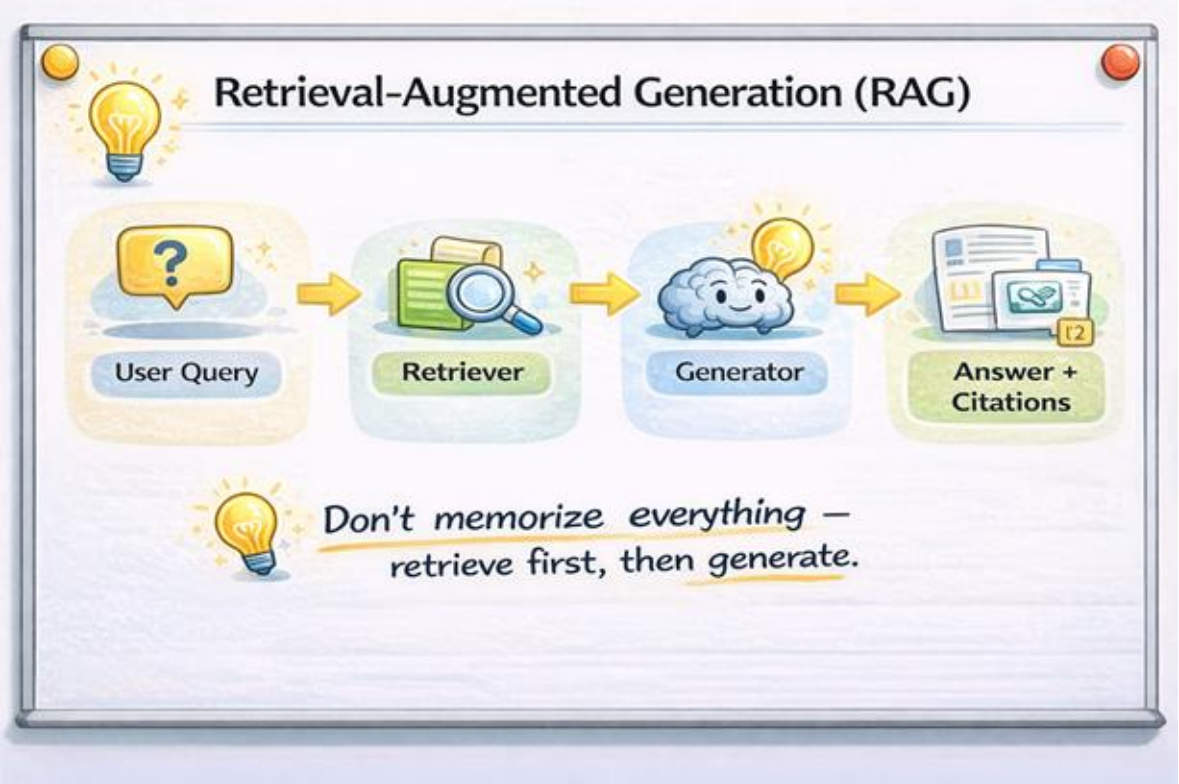
Why do we Need RAG?



Ways to get around the issues



What is RAG?


Retrieval-Augmented Generation (RAG)
User Query
Retriever
Generator
Answer + Citations
Don't memorize everything — retrieve first, then generate.

Example

You

I think Vin Diesel has been a voice actor for several characters in TV series, do you know what their names are? Please list all of them.

ChatGPT

Vin Diesel has lent his voice to several characters across various TV series and shows. Here's a list of the characters he's voiced:

1. **Groot** in "Guardians of the Galaxy Vol. 3" (2023 Movie), "The Guardians of the Galaxy Holiday Special" (2022 TV Show), "I Am Groot" (2022 Short), "Thor: Love and Thunder" (2022 Movie), "Avengers: Endgame" (2019 Movie), "Ralph Breaks the Internet: Wreck-It Ralph 2" (2018 Movie), "Avengers: Infinity War" (2018 Movie), "Guardians of the Galaxy Vol. 2" (2017 Movie), and "Guardians of the Galaxy" (2014 Movie) [↑].
2. **Santiago** in "ARK: The Animated Series" (2023 TV Show) [↑].
3. **Dominic Toretto** in "Fast & Furious: Spy Racers" (2019 TV Show) [↑].

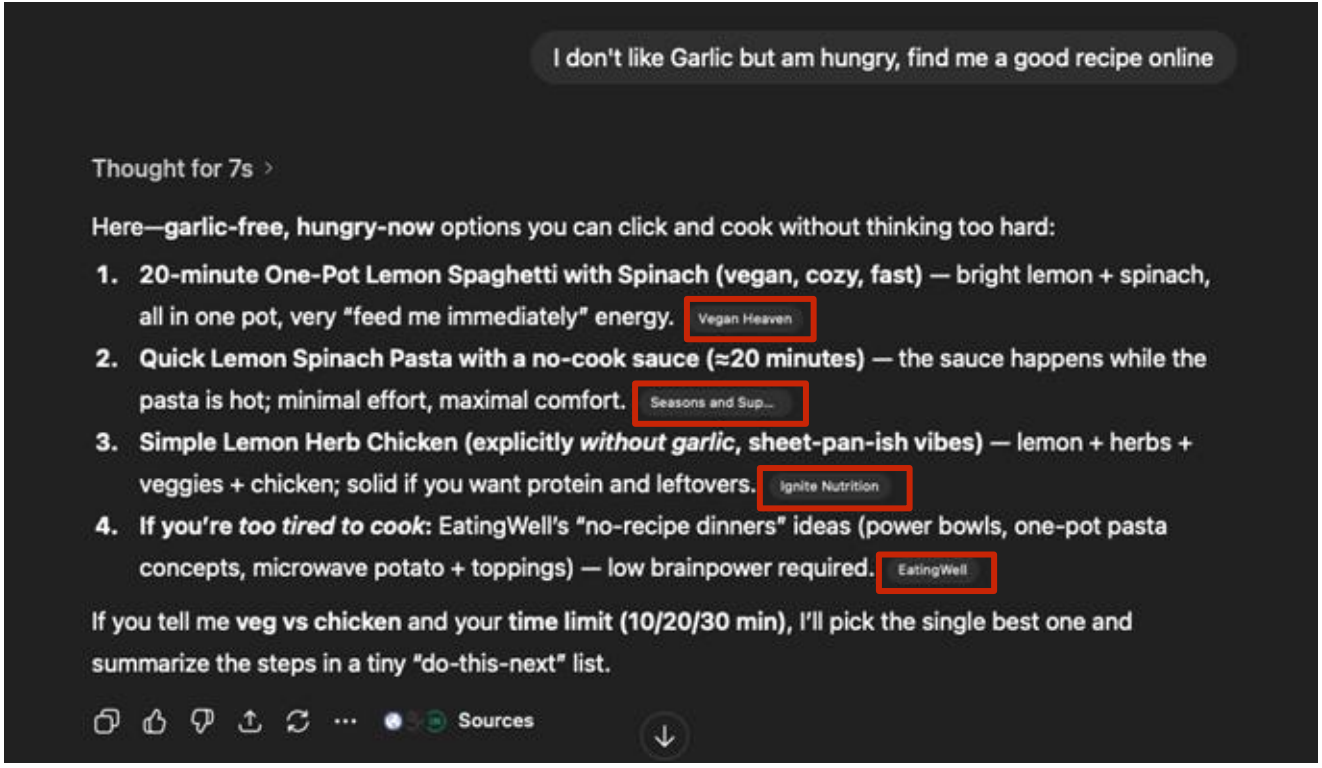
https://www.behindthevoiceactors.com/Vin-Diesel/

Additionally, though the website for "Big Mouth" lists Vin Diesel, it appears to be a misunderstanding or an error, as Nick Kroll is credited as the voice of Vin Diesel in that show [↑].

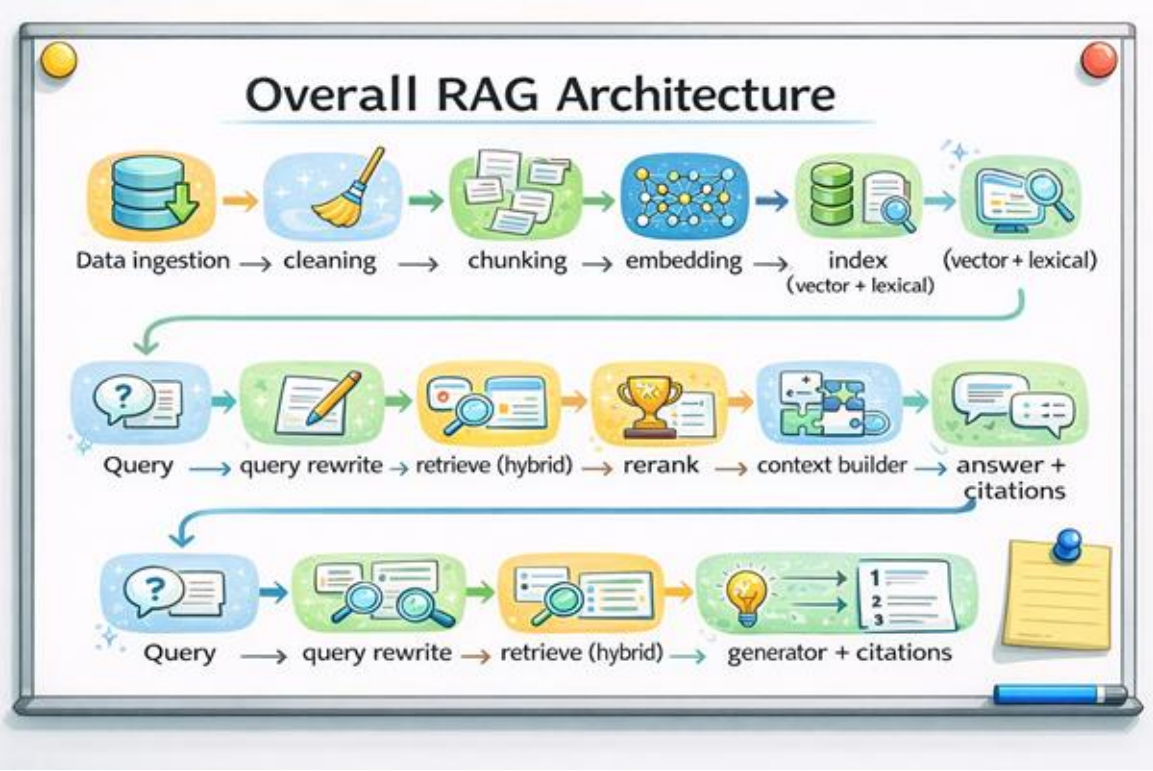
https://www.behindthevoiceactors.com/tv-shows/Big-Mouth/Vin-Diesel/

These roles showcase Diesel's versatility and contribution to voice acting, particularly in animated series and specials, where he brings iconic characters to life with his distinctive voice.

Example

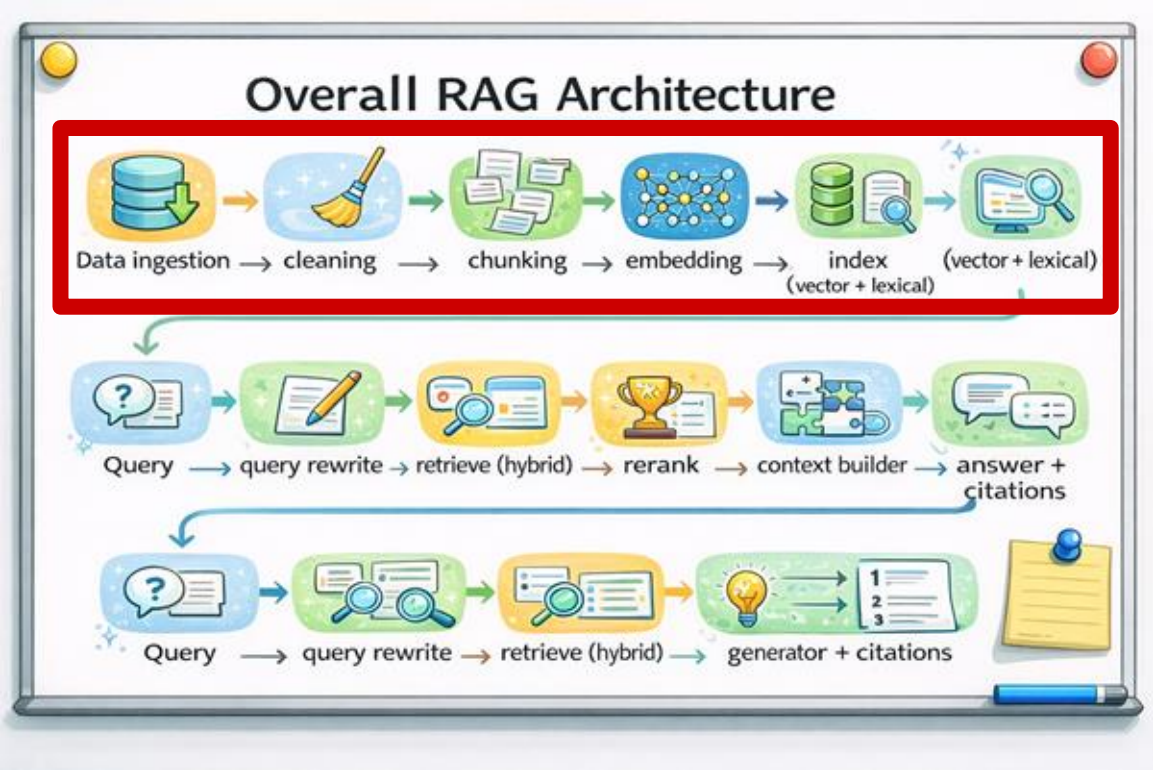

I don't like Garlic but am hungry, find me a good recipe online
Thought for 7s
Here—garlic-free, hungry-now options you can click and cook without thinking too hard:
1. 20-minute One-Pot Lemon Spaghetti with Spinach (vegan, cozy, fast) — bright lemon + spinach, all in one pot, very "feed me immediately" energy. Vegan Heaven
2. Quick Lemon Spinach Pasta with a no-cook sauce (≈20 minutes) — the sauce happens while the pasta is hot; minimal effort, maximal comfort. Seasons and Sup...
3. Simple Lemon Herb Chicken (explicitly without garlic, sheet-pan-ish vibes) — lemon + herbs + veggies + chicken; solid if you want protein and leftovers. Ignite Nutrition
4. If you're too tired to cook: EatingWell's "no-recipe dinners" ideas (power bowls, one-pot pasta concepts, microwave potato + toppings) — low brainpower required. EatingWell
If you tell me veg vs chicken and your time limit (10/20/30 min), I'll pick the single best one and summarize the steps in a tiny "do-this-next" list.
Sources

RAG Architecture

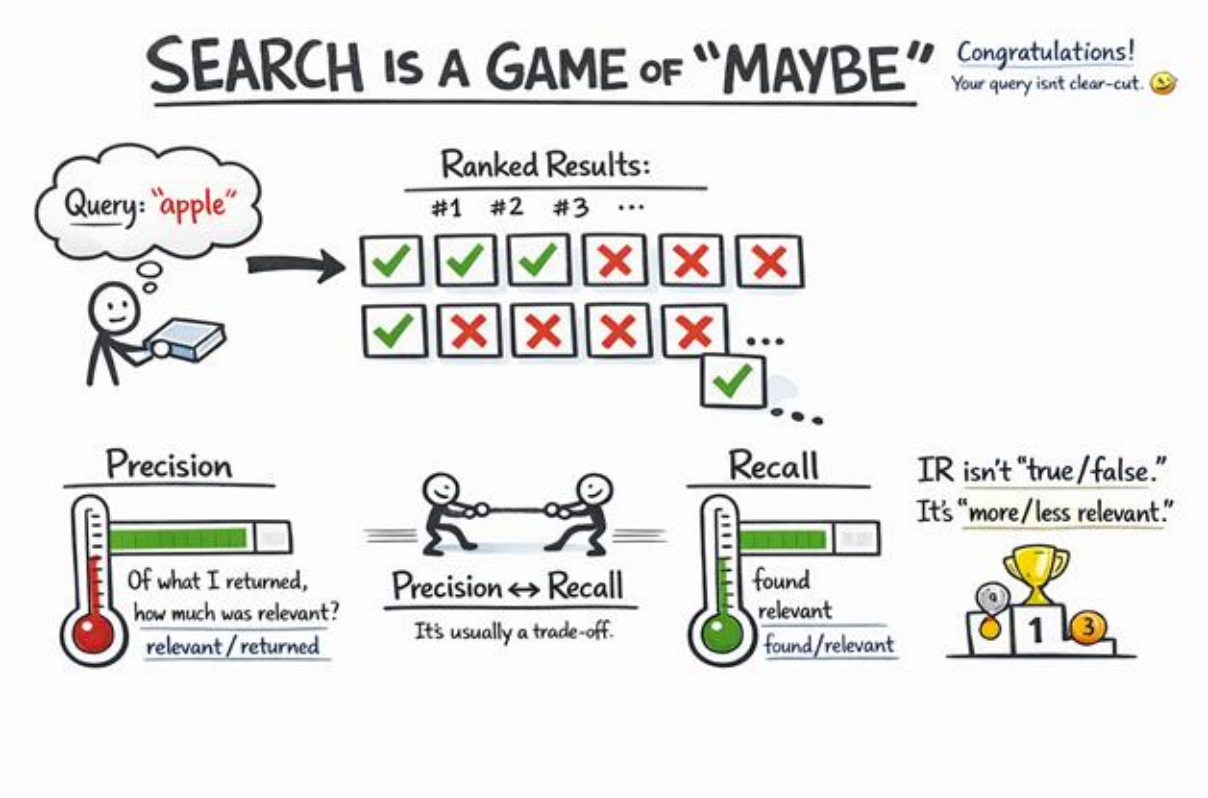


Retrieval Methods

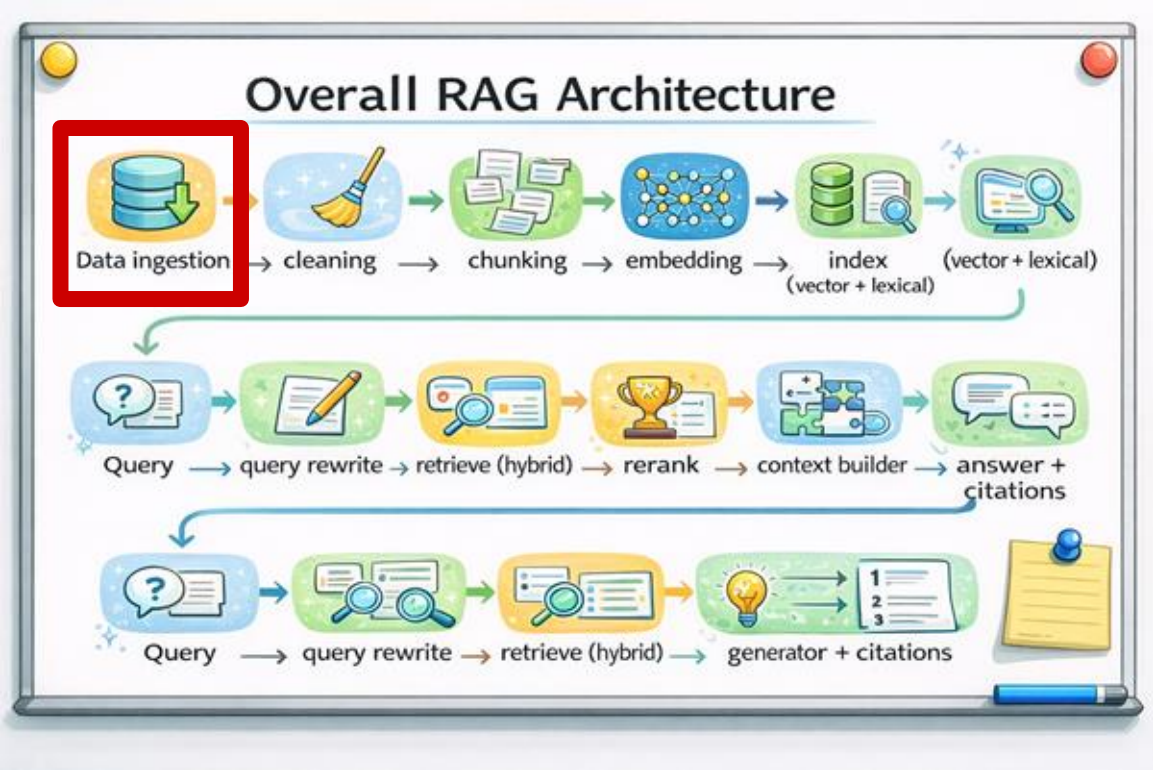
RAG Architecture



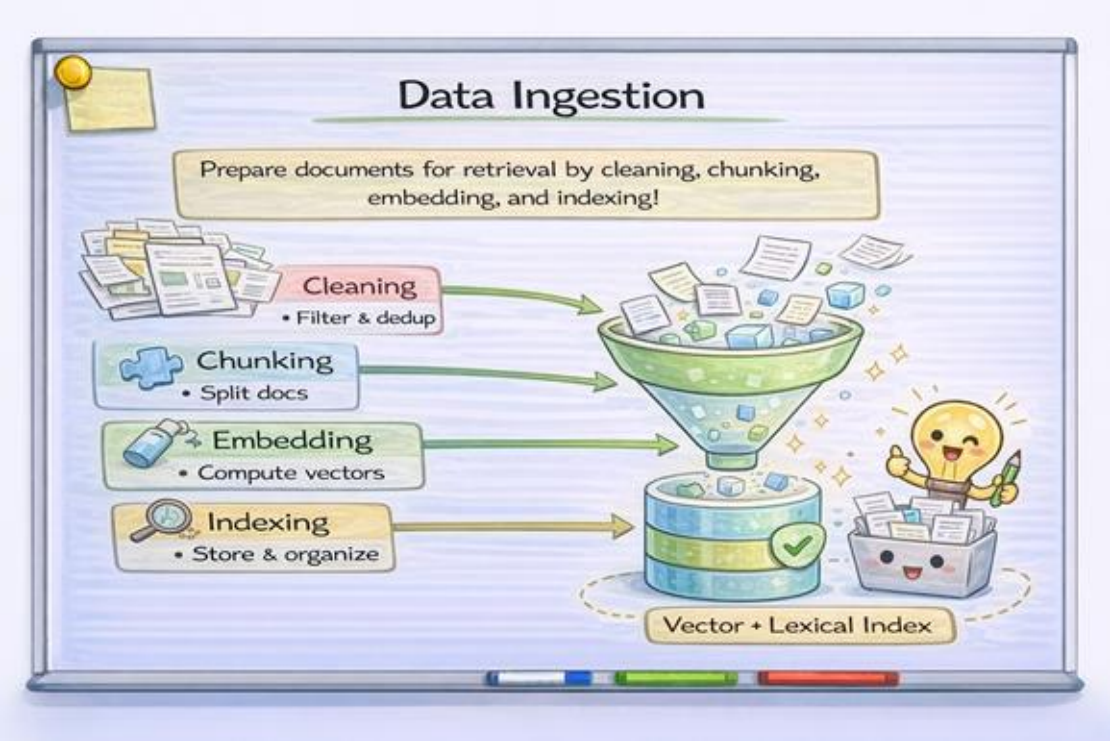
What is Information Retrival?



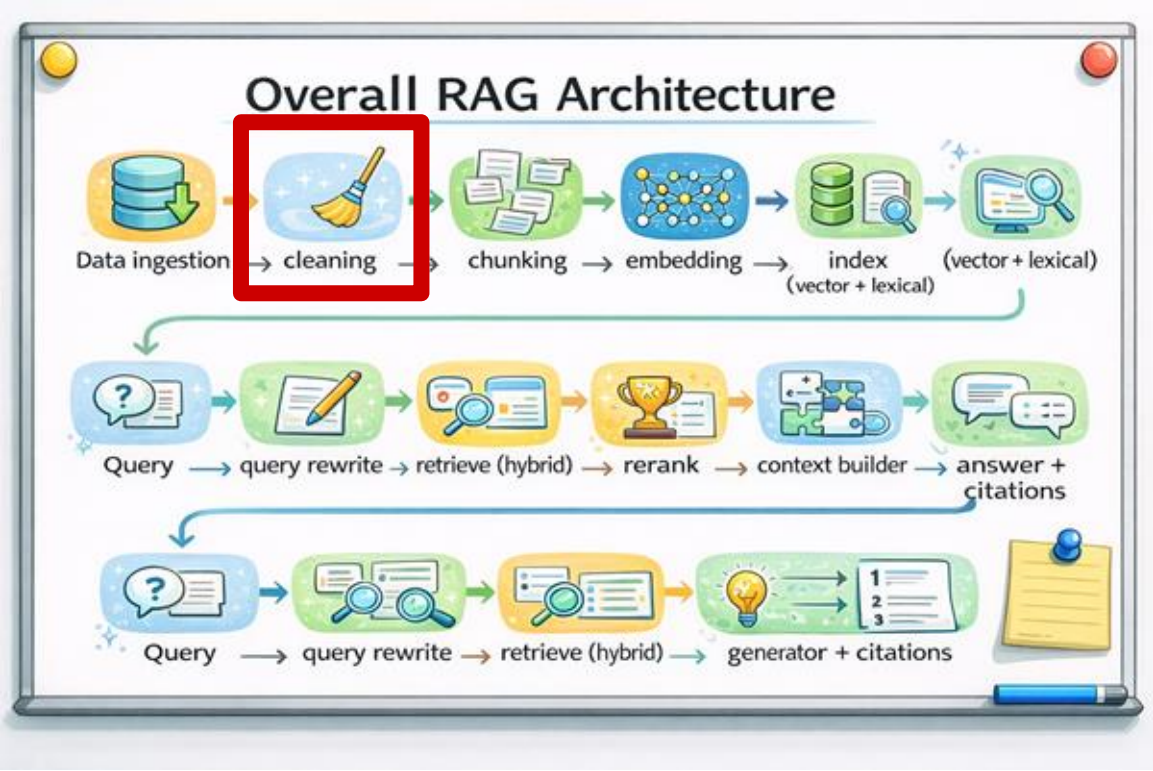
RAG Architecture



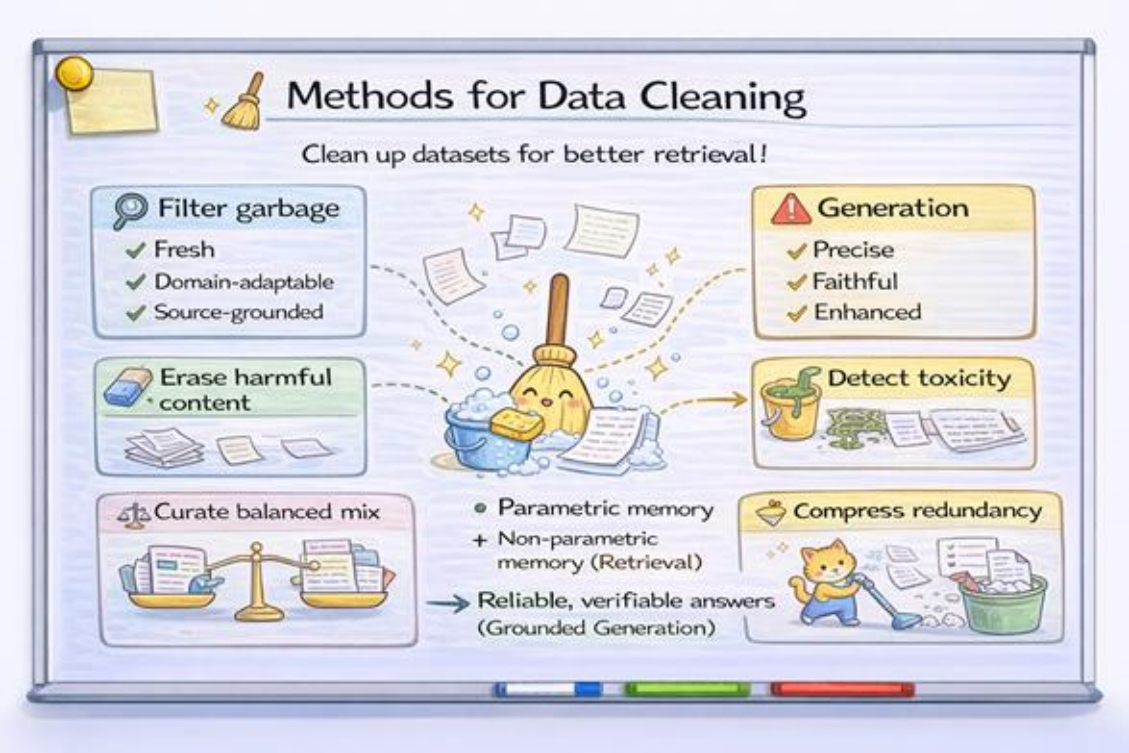
Data Ingestion



RAG Architecture



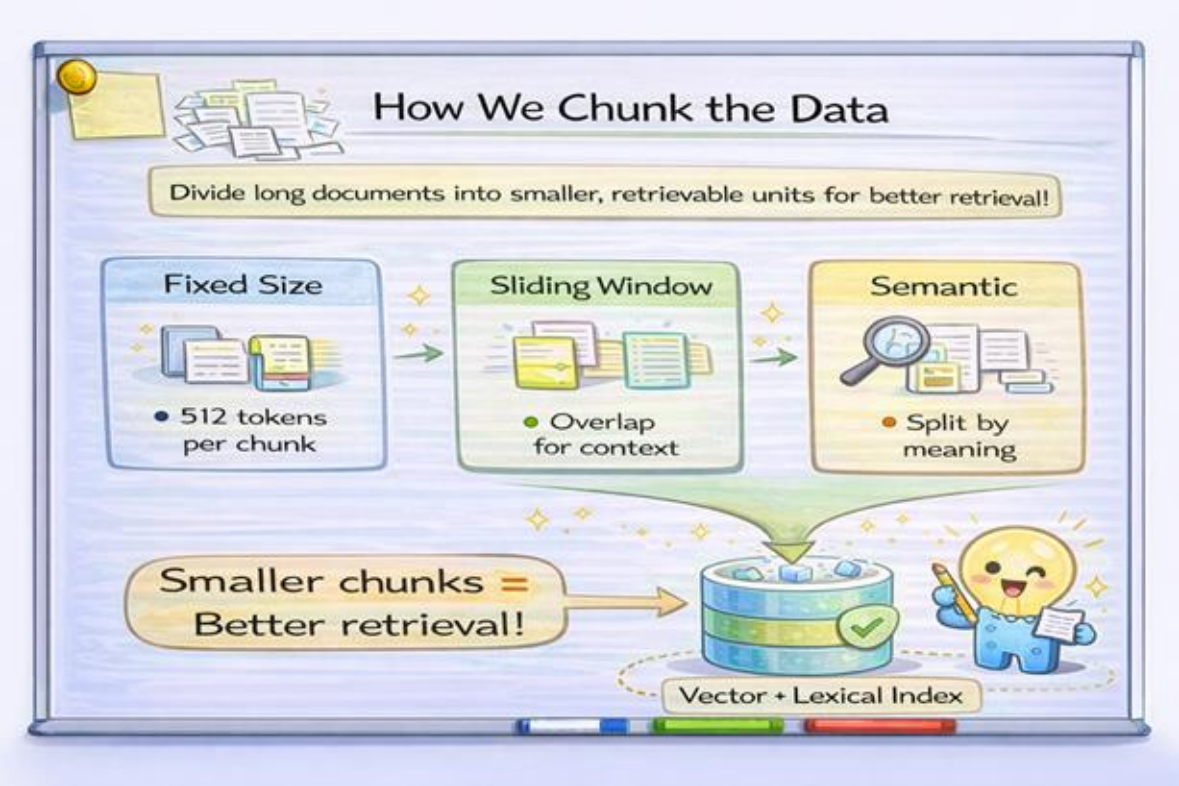
Let’s do some cleaning…



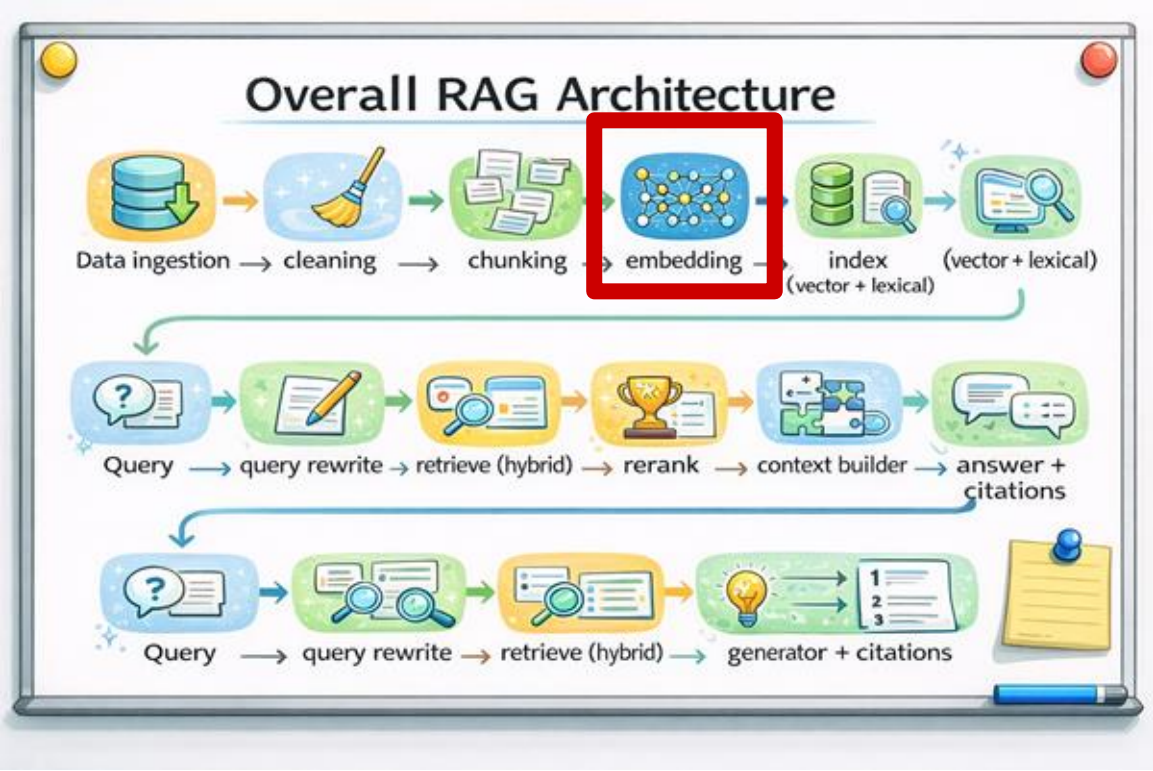
RAG Architecture



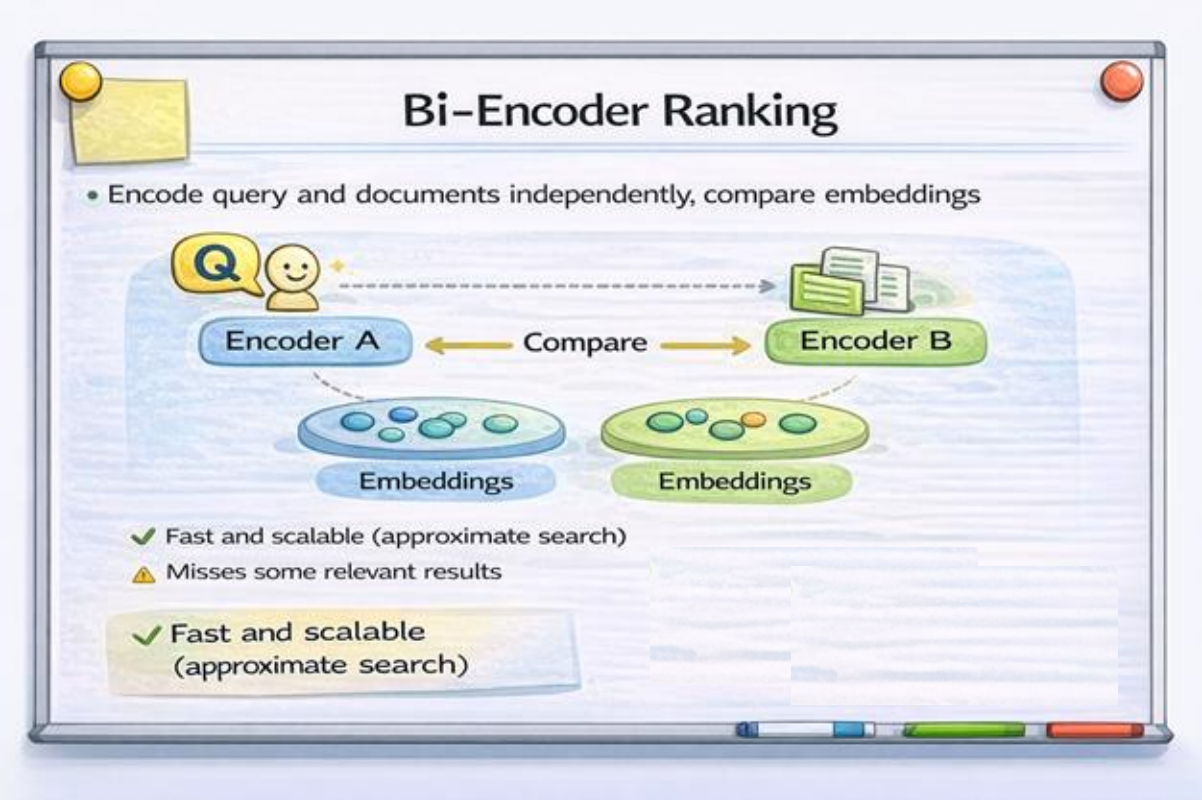
Chunking


How We Chunk the Data
Divide long documents into smaller, retrievable units for better retrieval!
Fixed Size
512 tokens per chunk
Sliding Window
Overlap for context
Semantic
Split by meaning
Smaller chunks = Better retrieval!
Vector + Lexical Index

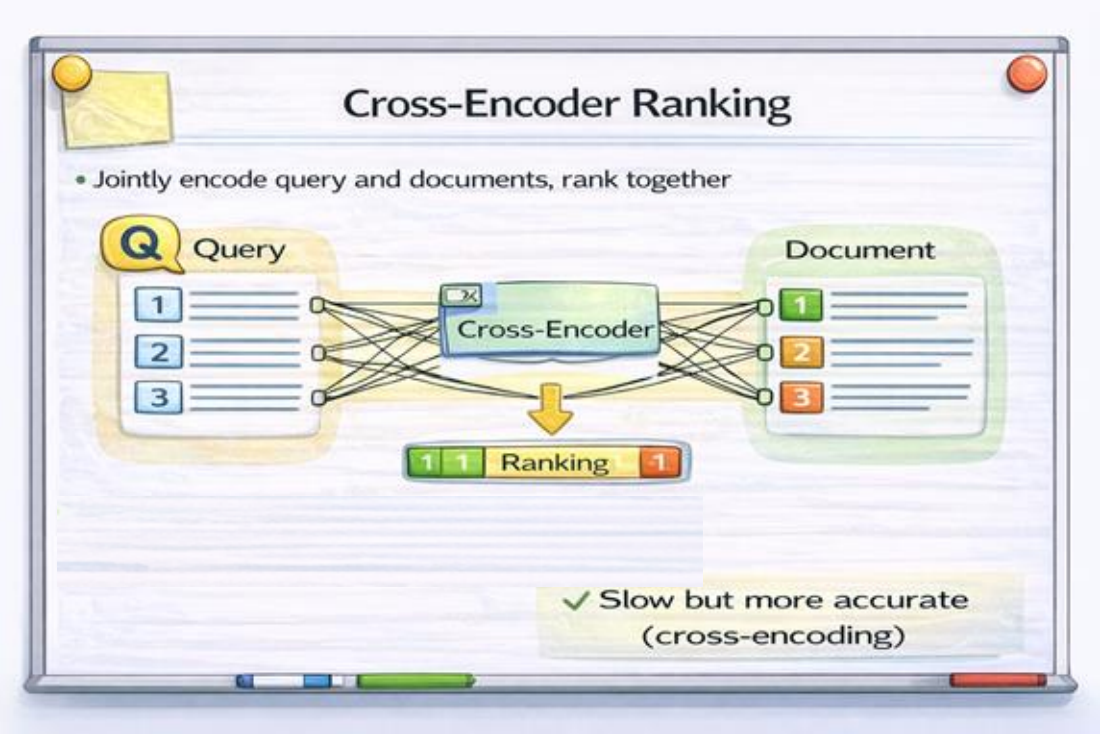
RAG Architecture



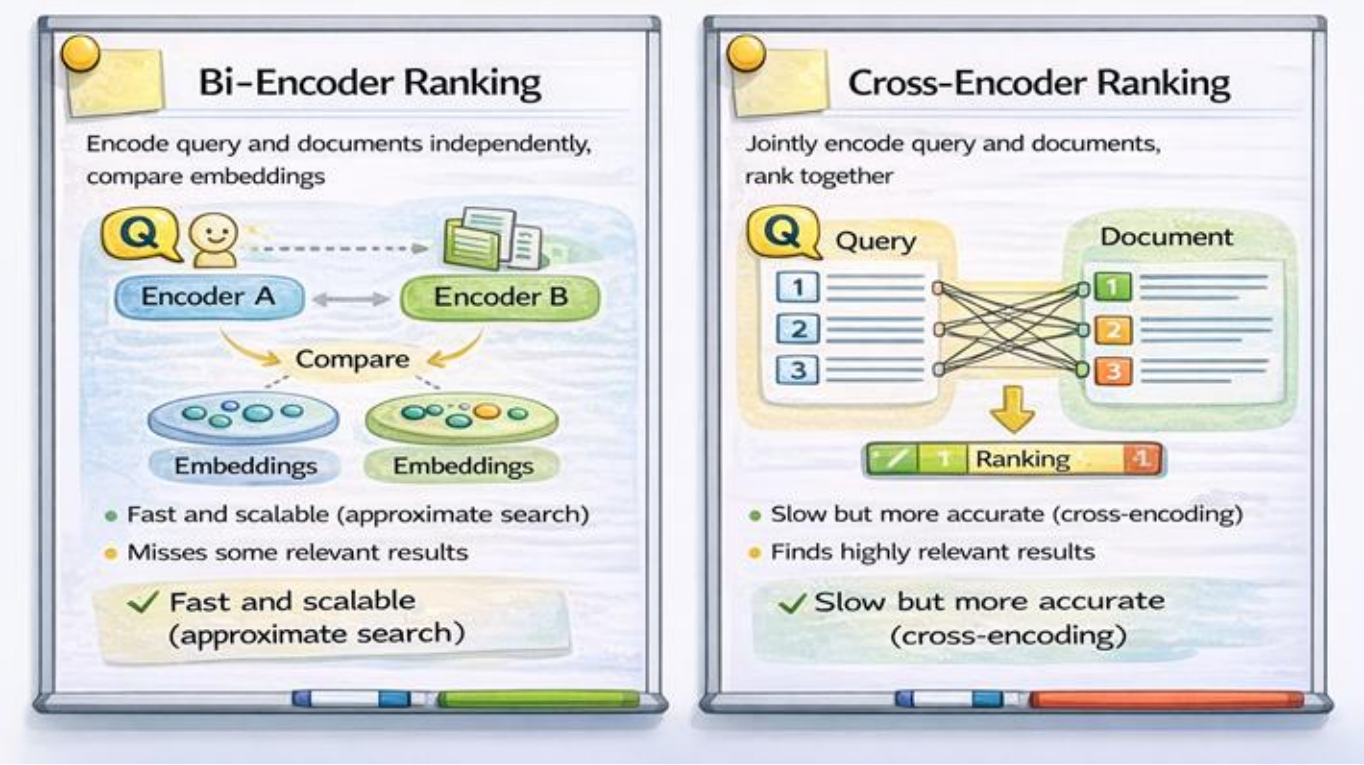
Encoder Ranking



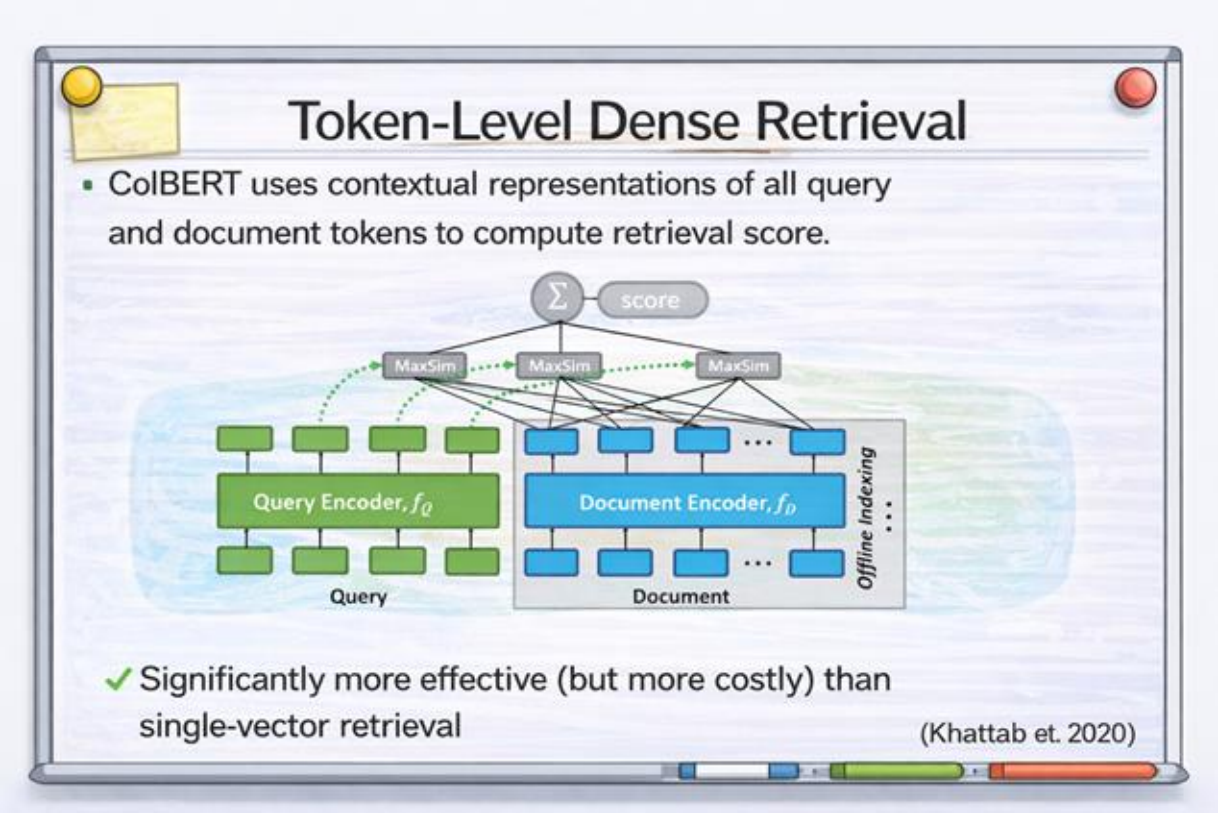
Encoder Ranking



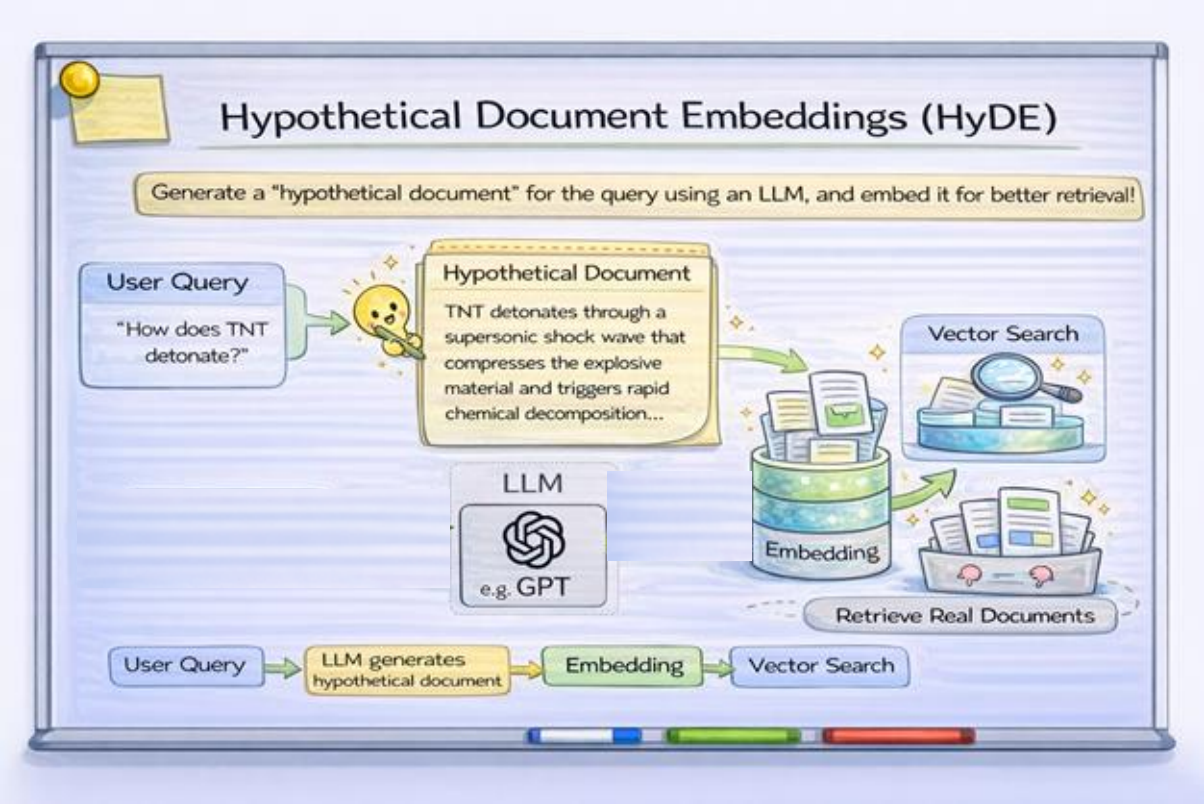
Encoder Ranking



Token-level Dens Retrival

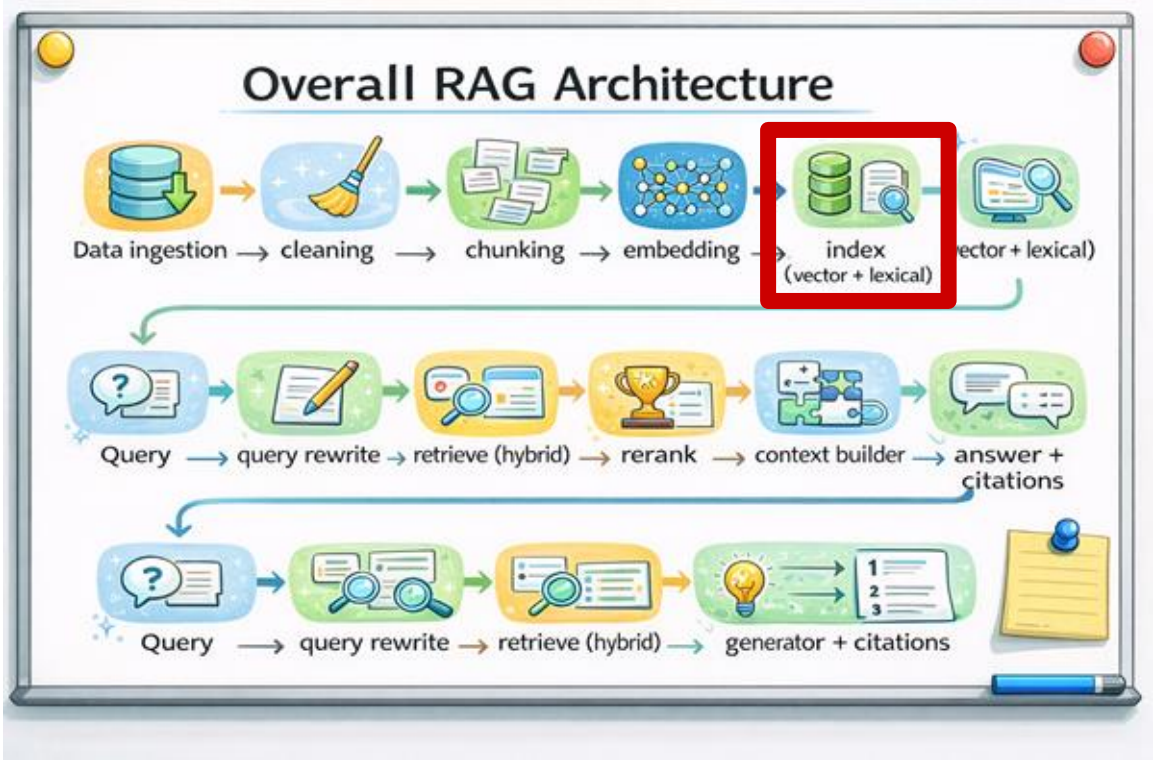


Hypothetical Document Embeddings

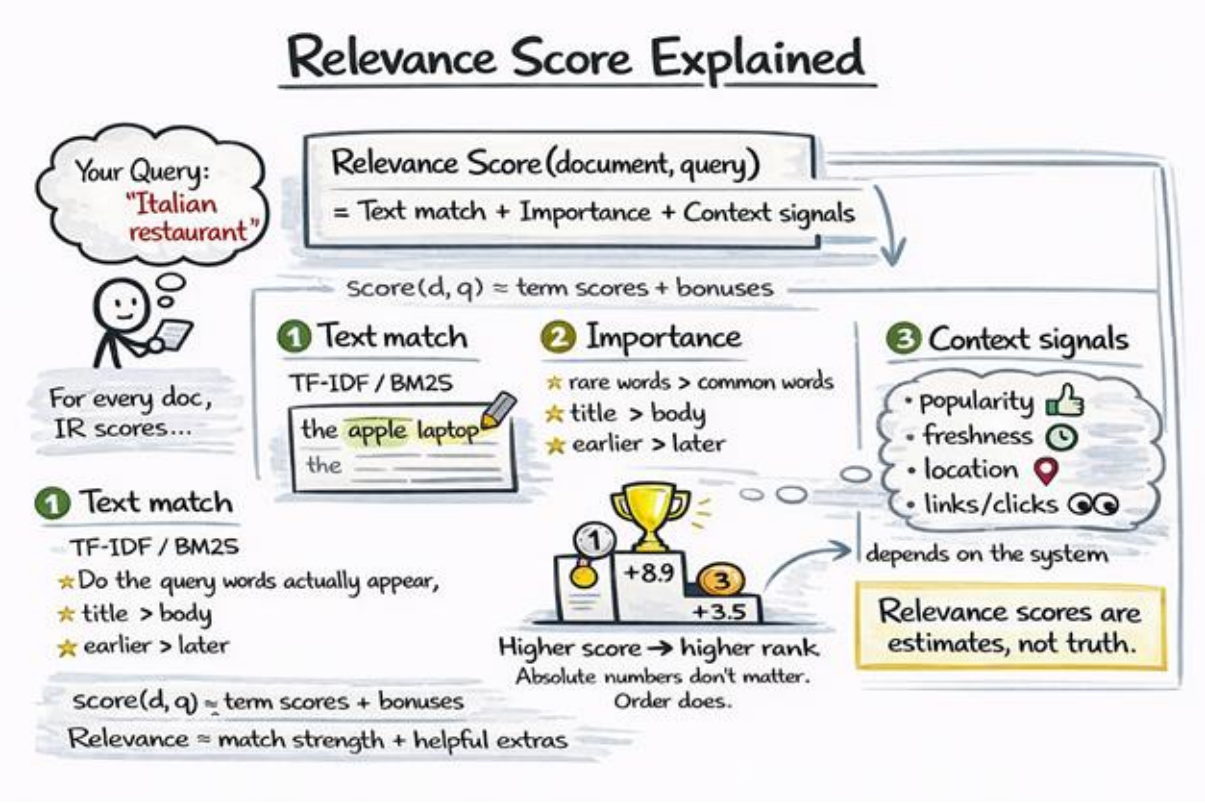


(Gao et al. 2022)

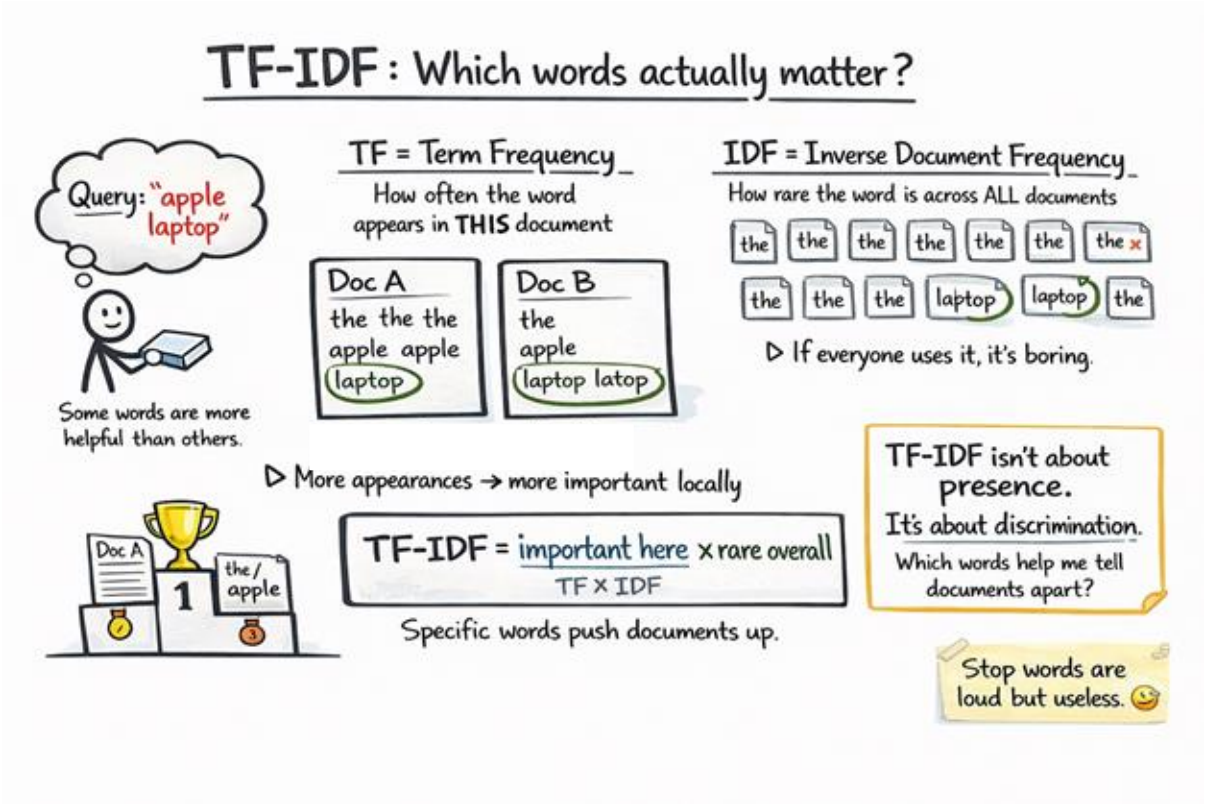
RAG Architecture



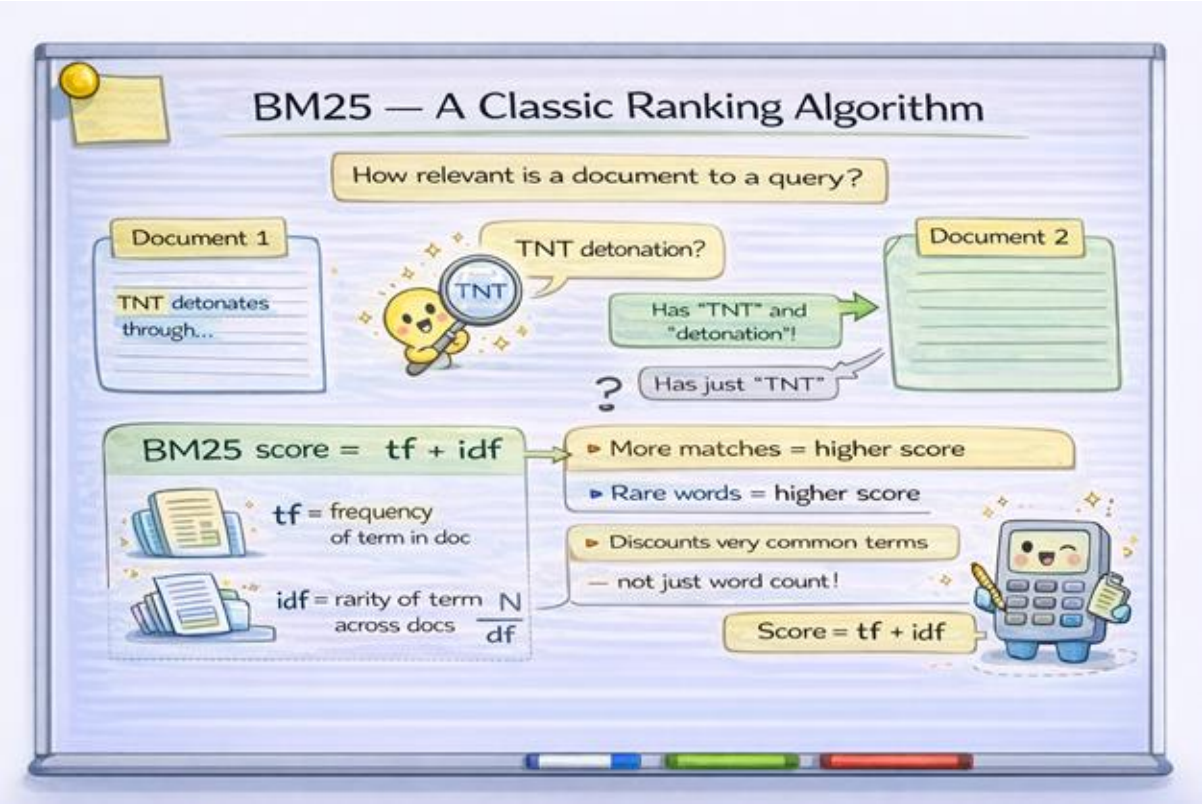
Relevance score



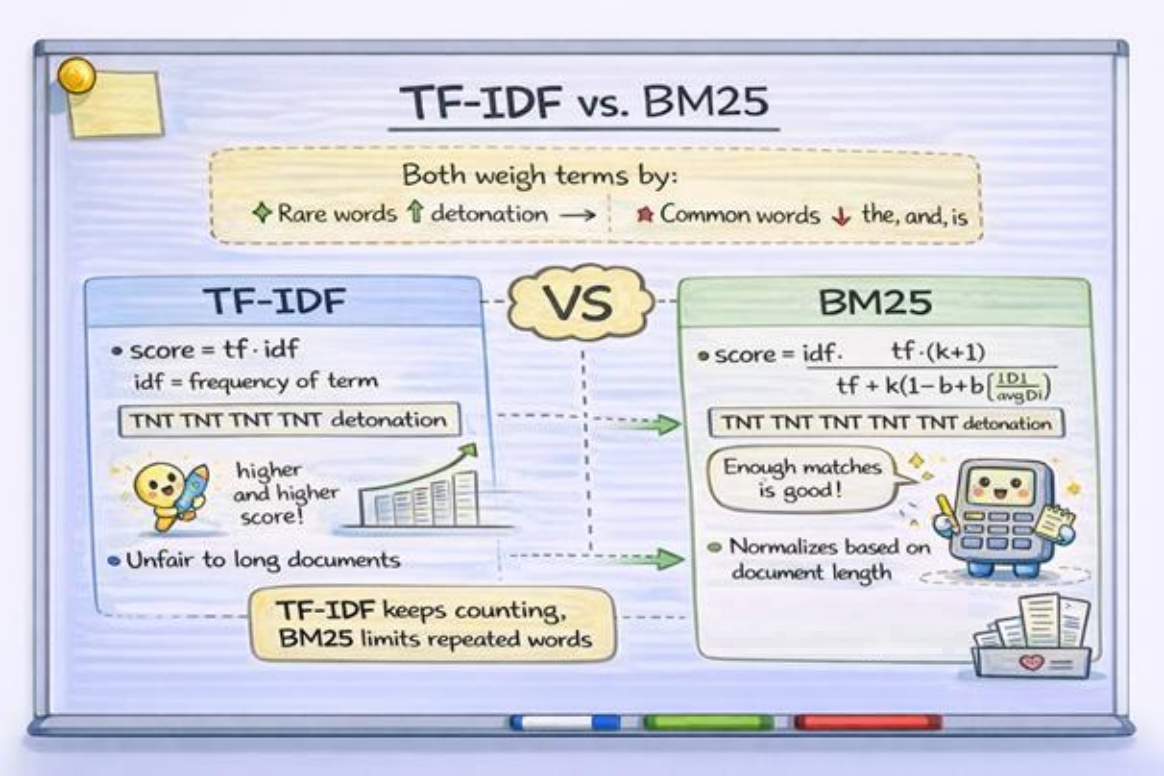
TD-IDF


TF-IDF : Which words actually matter?
Query: "apple laptop"
Some words are more helpful than others.
TF = Term Frequency
How often the word appears in THIS document
Doc A
the the the apple apple laptop
Doc B
the apple laptop latop
IDF = Inverse Document Frequency
How rare the word is across ALL documents
the the the the the the the x
the the the laptop laptop the
▷ If everyone uses it, it's boring.
▷ More appearances → more important locally
TF-IDF = important here x rare overall
TF x IDF
Specific words push documents up.
Doc A
1
the / apple
TF-IDF isn't about presence.
It's about discrimination.
Which words help me tell documents apart?
Stop words are loud but useless.

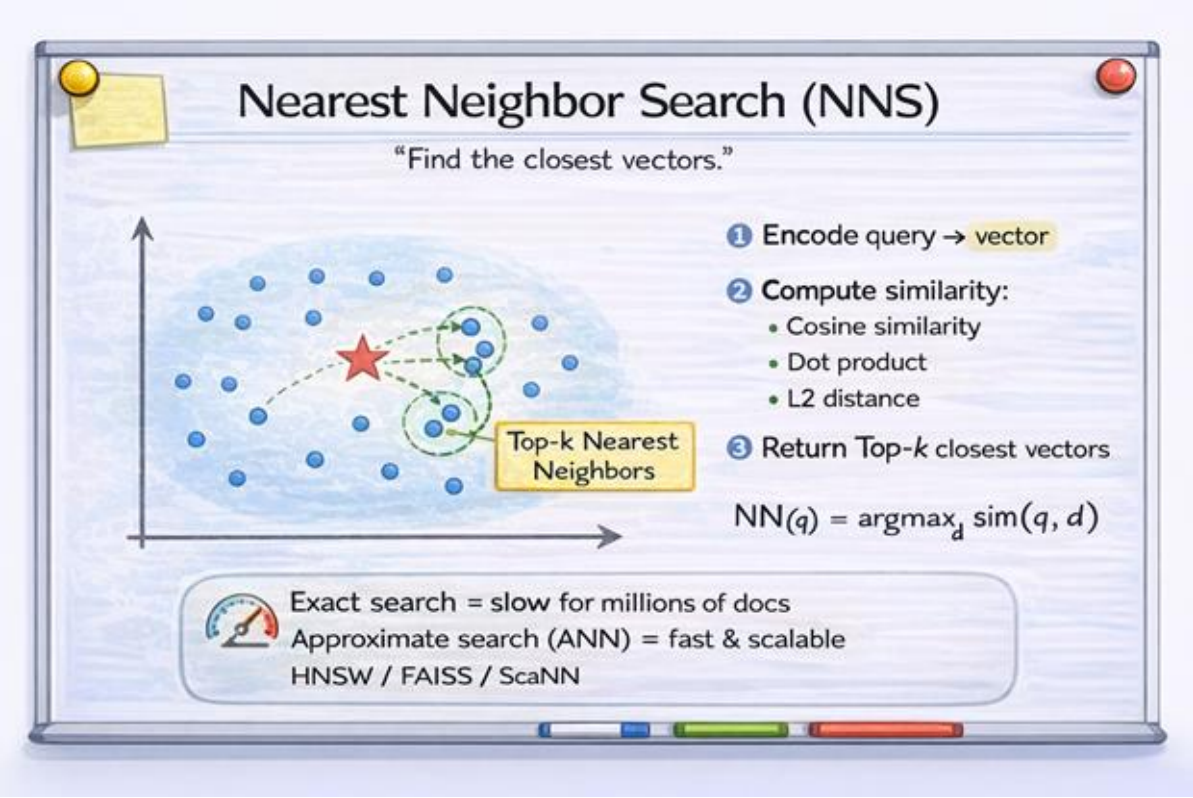
BM25


BM25 — A Classic Ranking Algorithm
How relevant is a document to a query?
Document 1
TNT detonates through...
TNT
TNT detonation?
Has "TNT" and "detonation"!
Has just "TNT"
Document 2
BM25 score = tf + idf
tf = frequency of term in doc
idf = rarity of term across docs
N
df
More matches = higher score
Rare words = higher score
Discounts very common terms
— not just word count!
Score = tf + idf

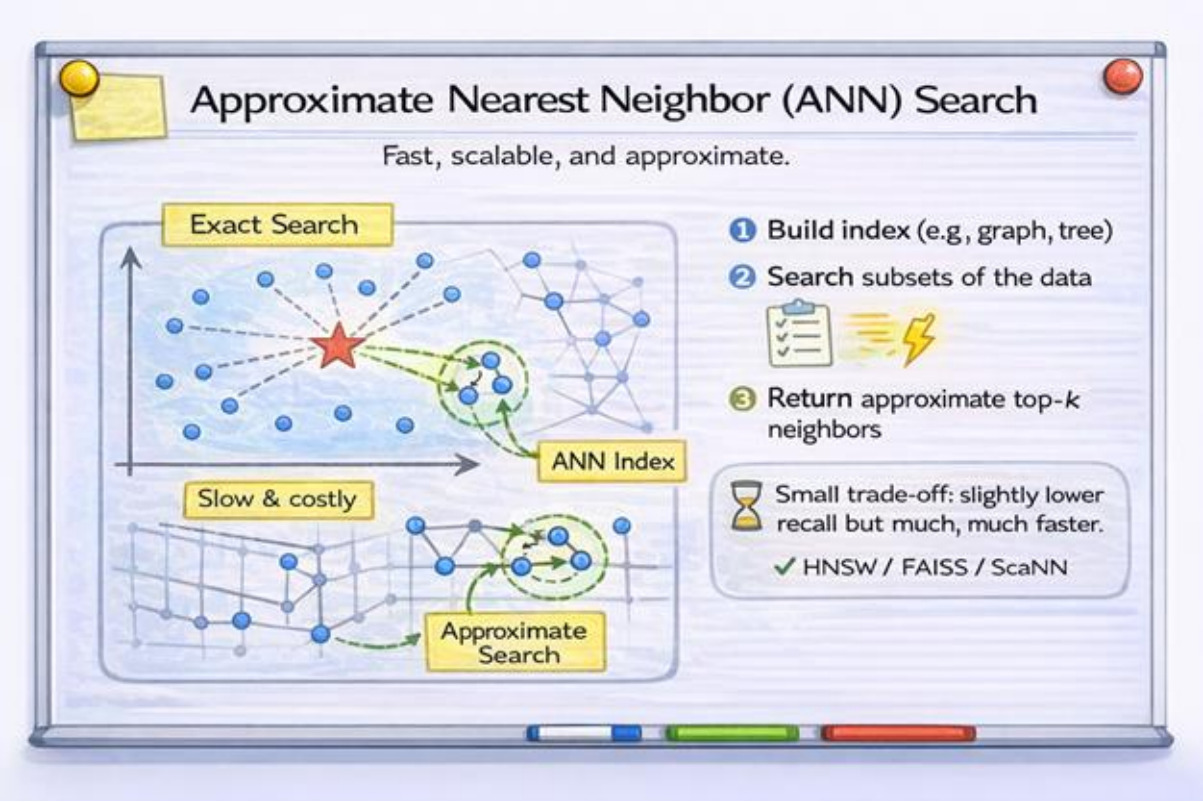
TD-IDF vs. BM25



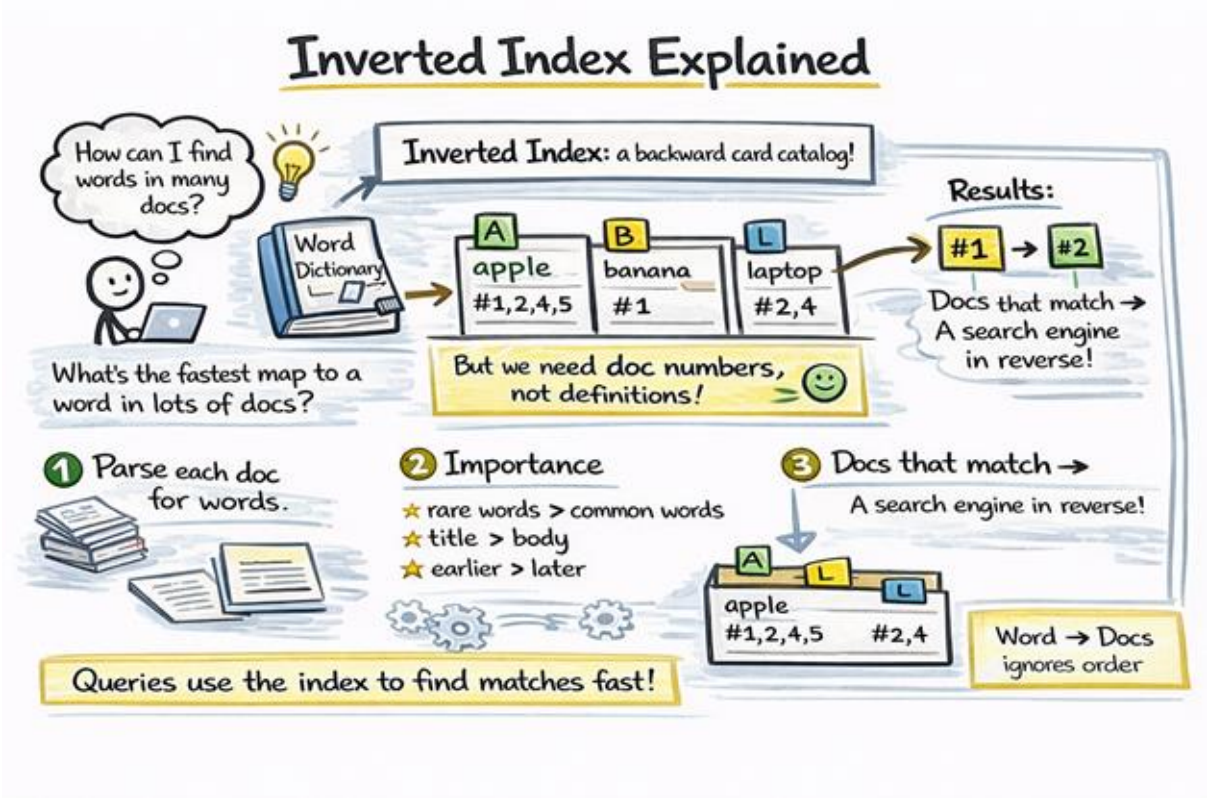
Nearest Neighbor Search



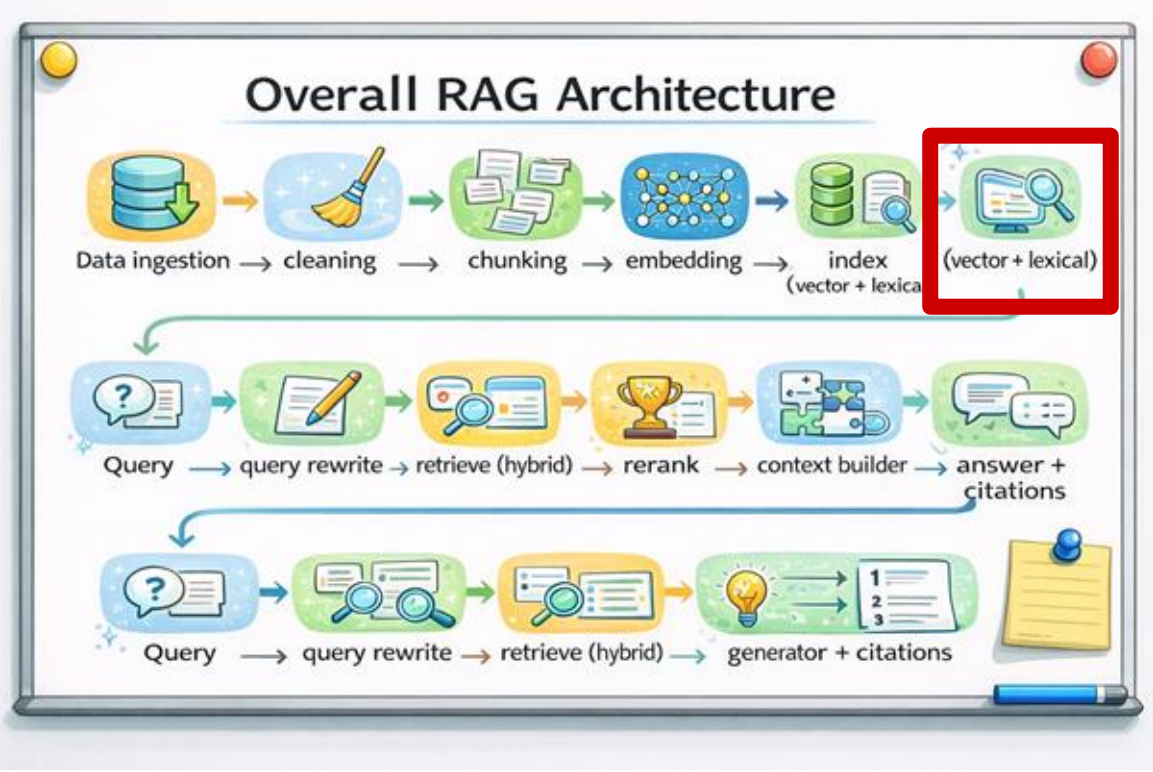
Nearest Neighbor Search



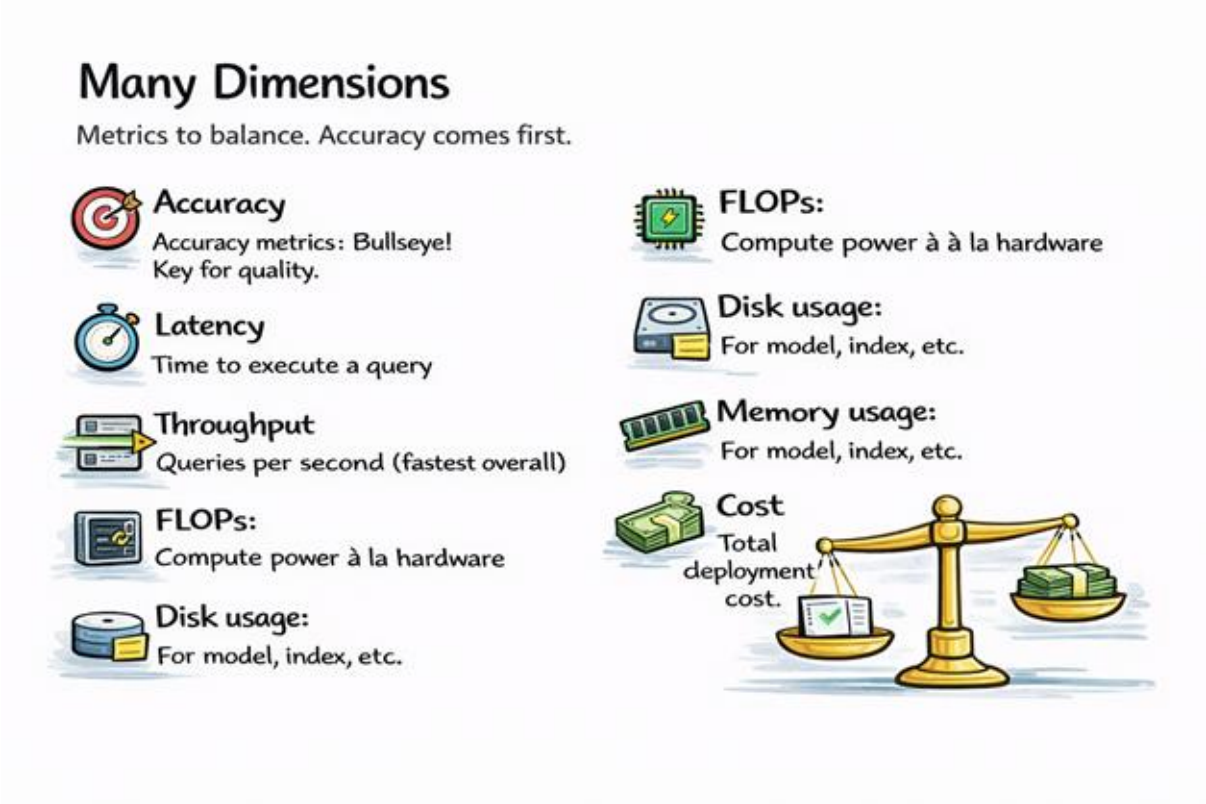
Inverted Indices



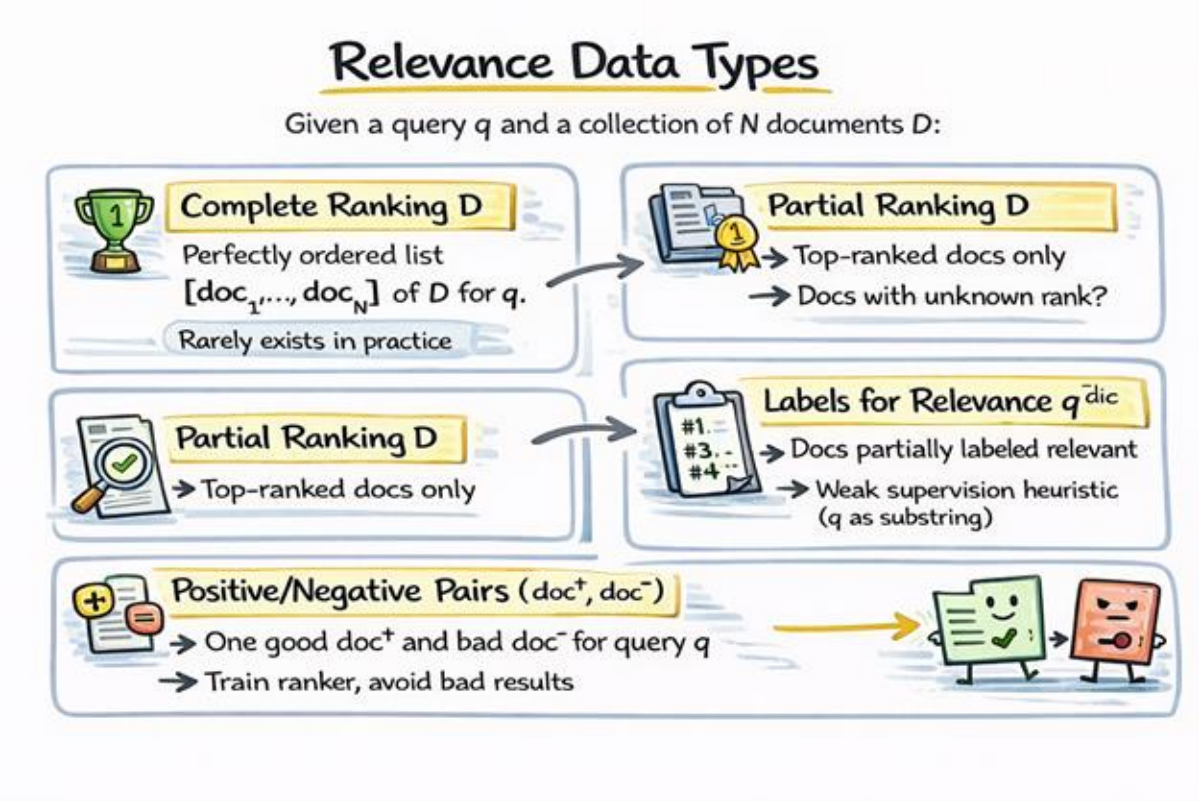
RAG Architecture



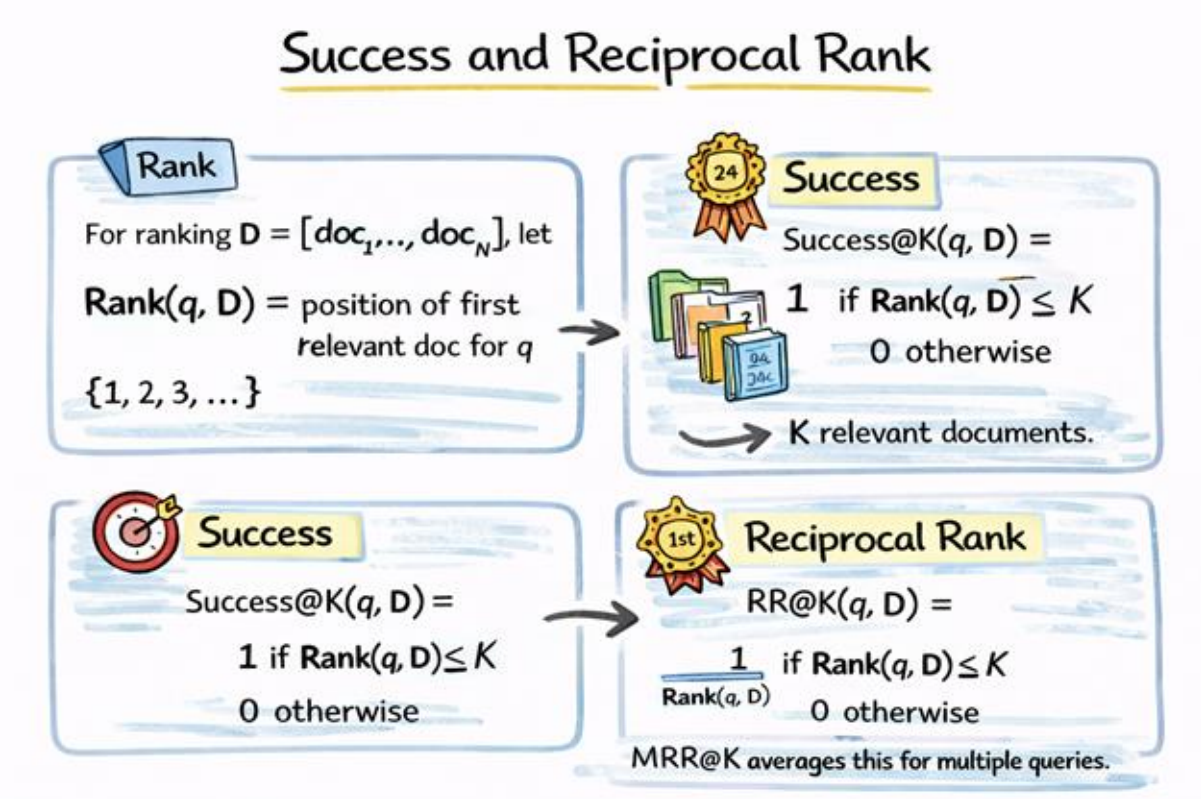
Dimensions


Many Dimensions
Metrics to balance. Accuracy comes first.
Accuracy
Accuracy metrics: Bullseye!
Key for quality.
Latency
Time to execute a query
Throughput
Queries per second (fastest overall)
FLOPs:
Compute power à la hardware
Disk usage:
For model, index, etc.
FLOPs:
Compute power à à la hardware
Disk usage:
For model, index, etc.
Memory usage:
For model, index, etc.
Cost
Total deployment cost.

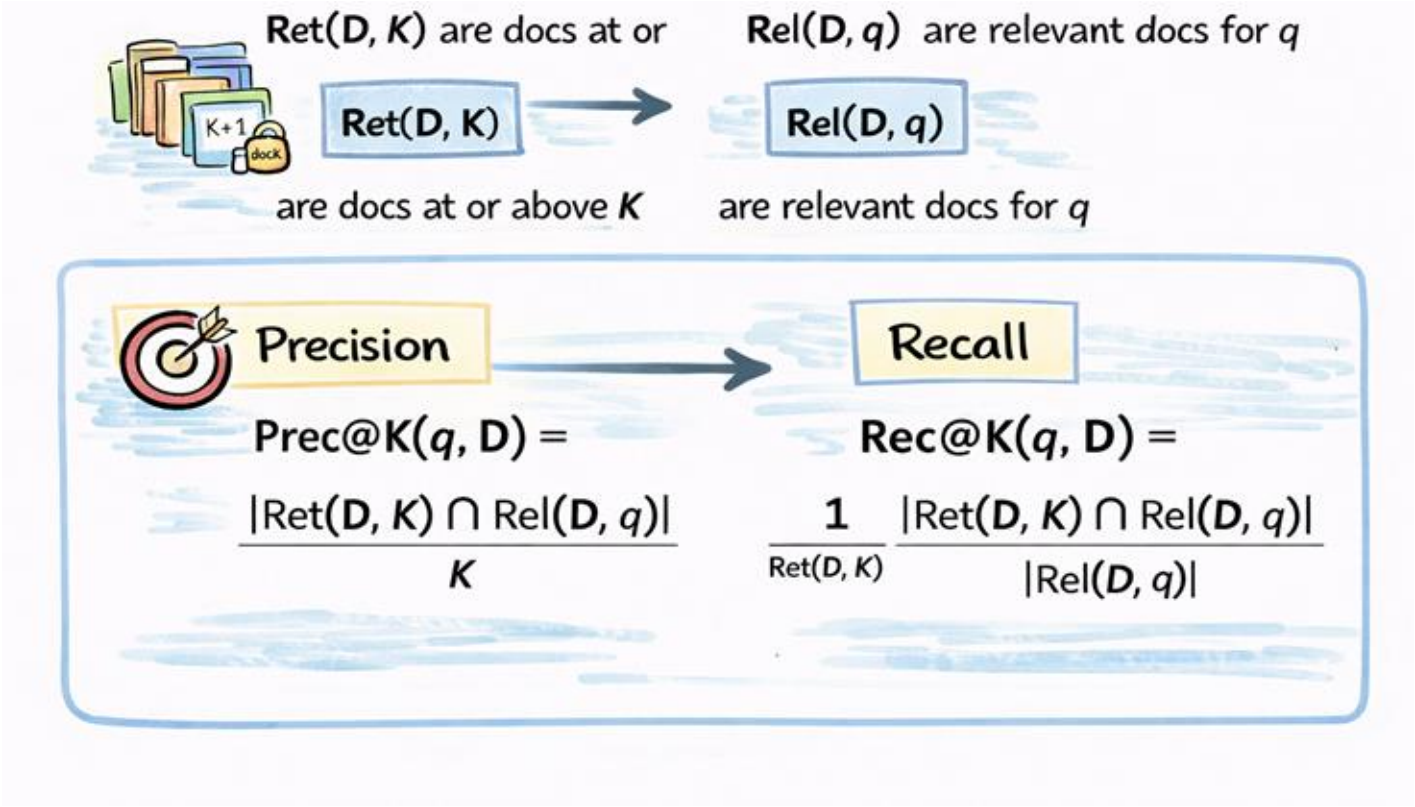
Relevance Data types



Success and Reciprocal Rank



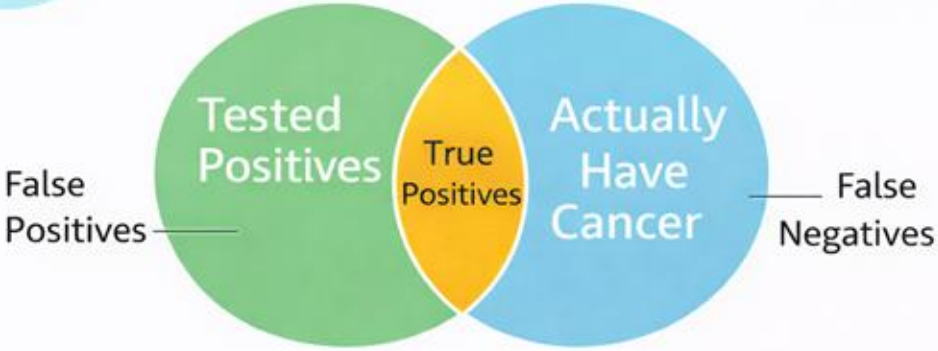
Precision and Recall



Precision and Recall: An Example



People who test positive for cancer vs. people who actually have cancer



Precision measures correctness.

$$\text{Precision} = \frac{\#\text{ True Positives}}{\#\text{ Tested Positive}}$$

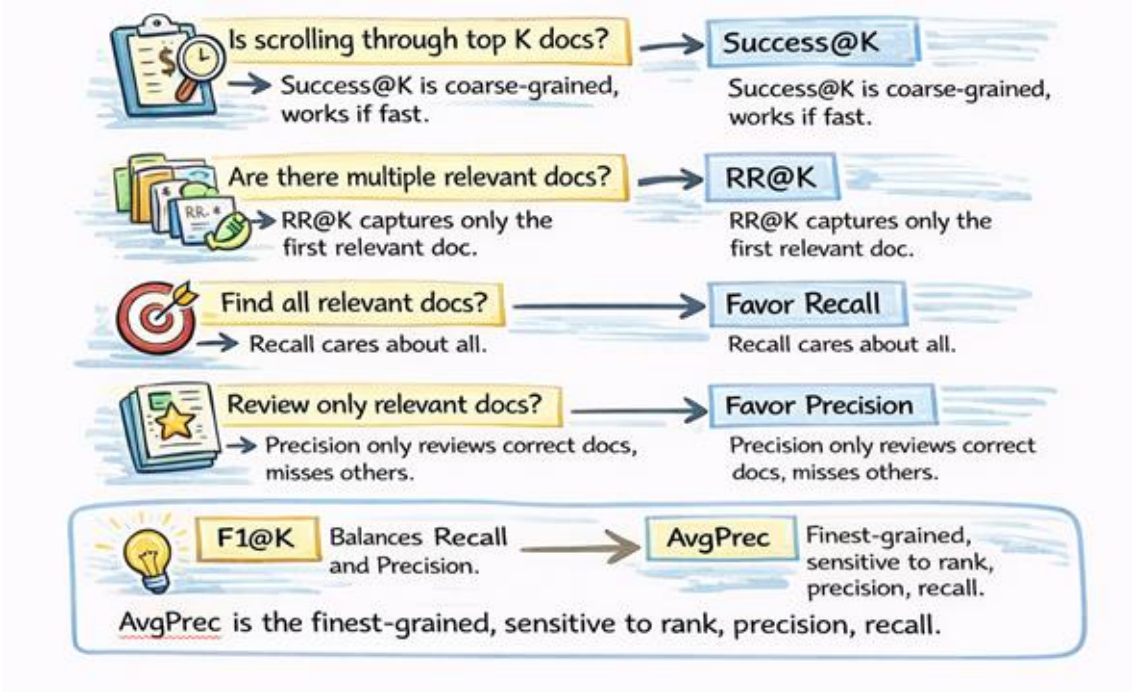
How many testing positive actually have cancer?

Recall measures coverage

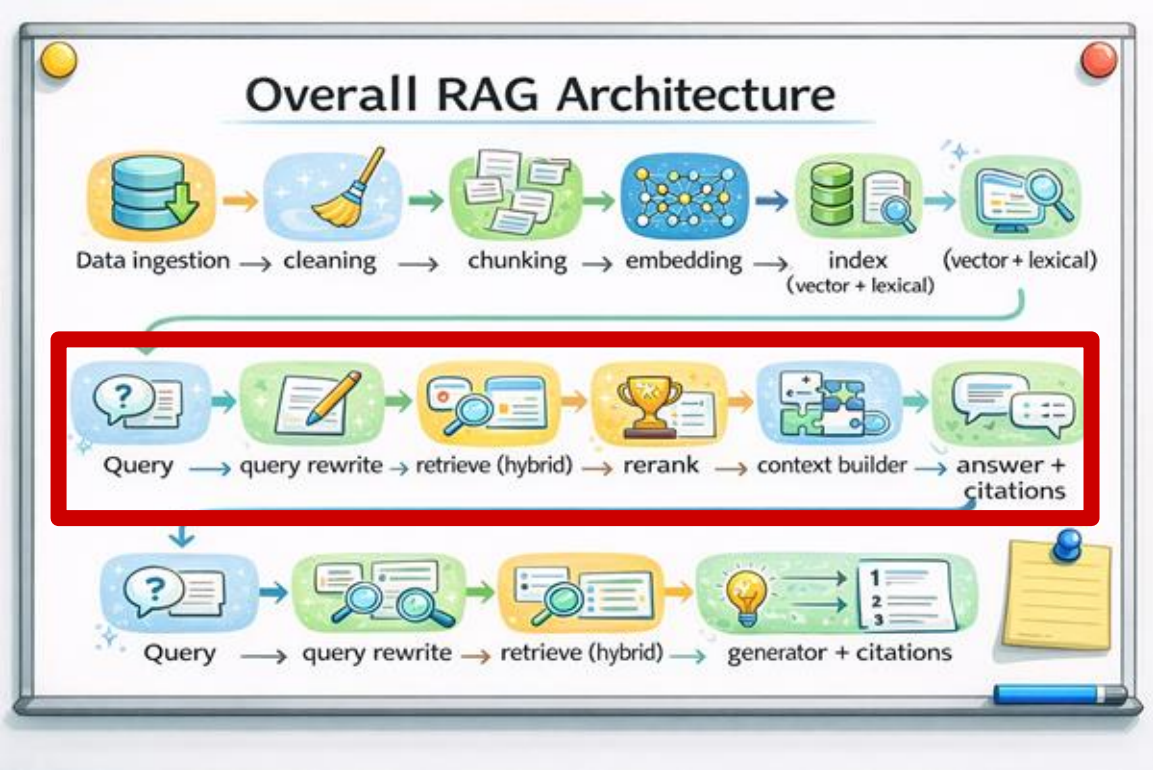
$$\text{Recall} = \frac{\#\text{ True Positives}}{\#\text{ Actually Have Cancer}}$$

How many with cancer were picked up by the test?

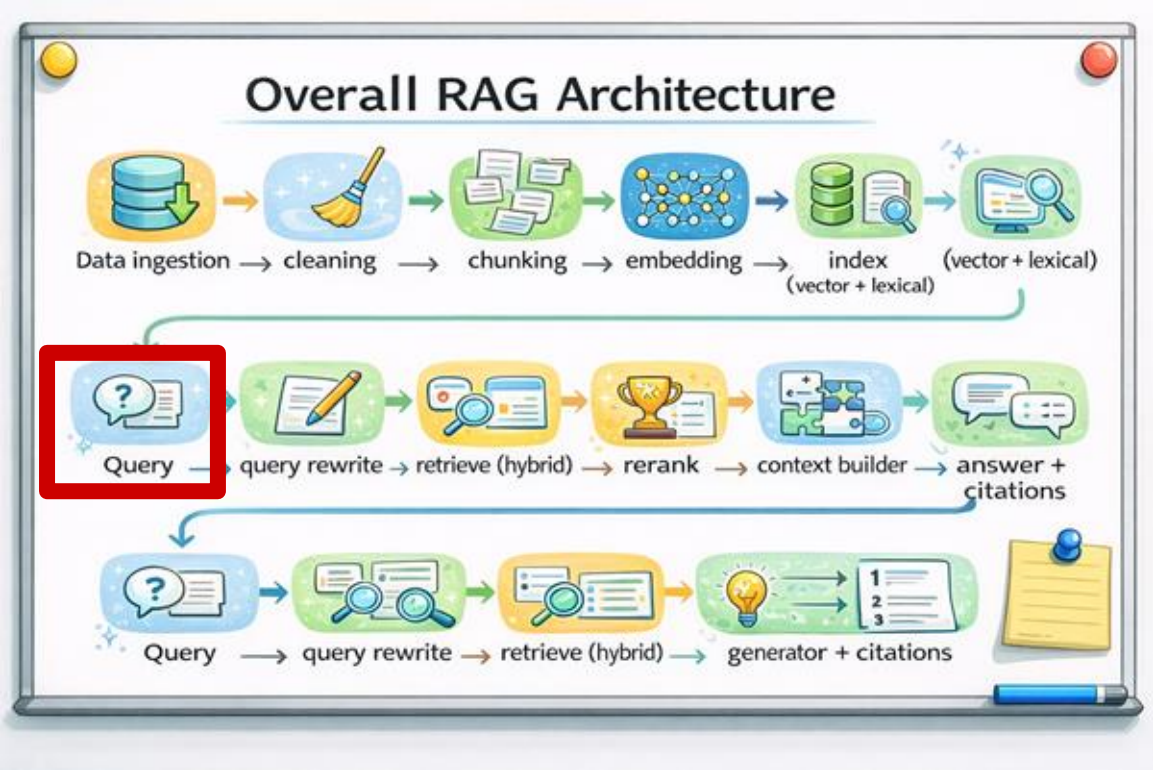
Which metric? There is no single answer!



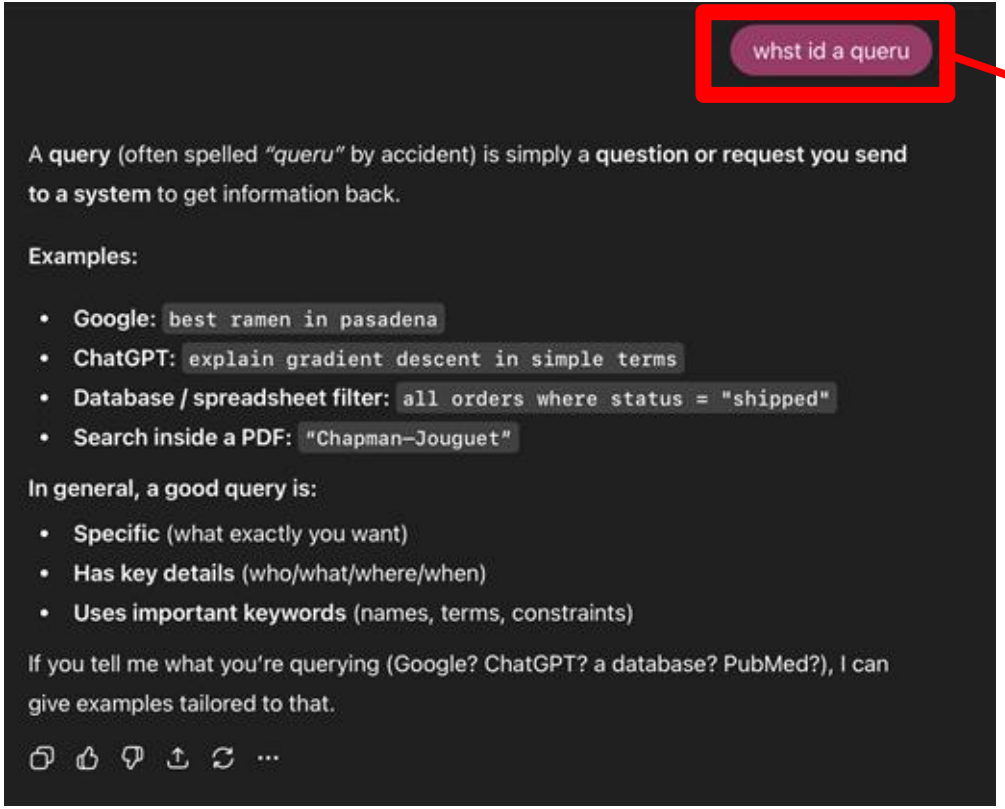
Prompting


Overall RAG Architecture
Data ingestion
cleaning
chunking
embedding
index
(vector + lexical)
(vector + lexical)
Query
query rewrite
retrieve (hybrid)
rerank
context builder
answer + citations
Query
query rewrite
retrieve (hybrid)
generator + citations

Prompting

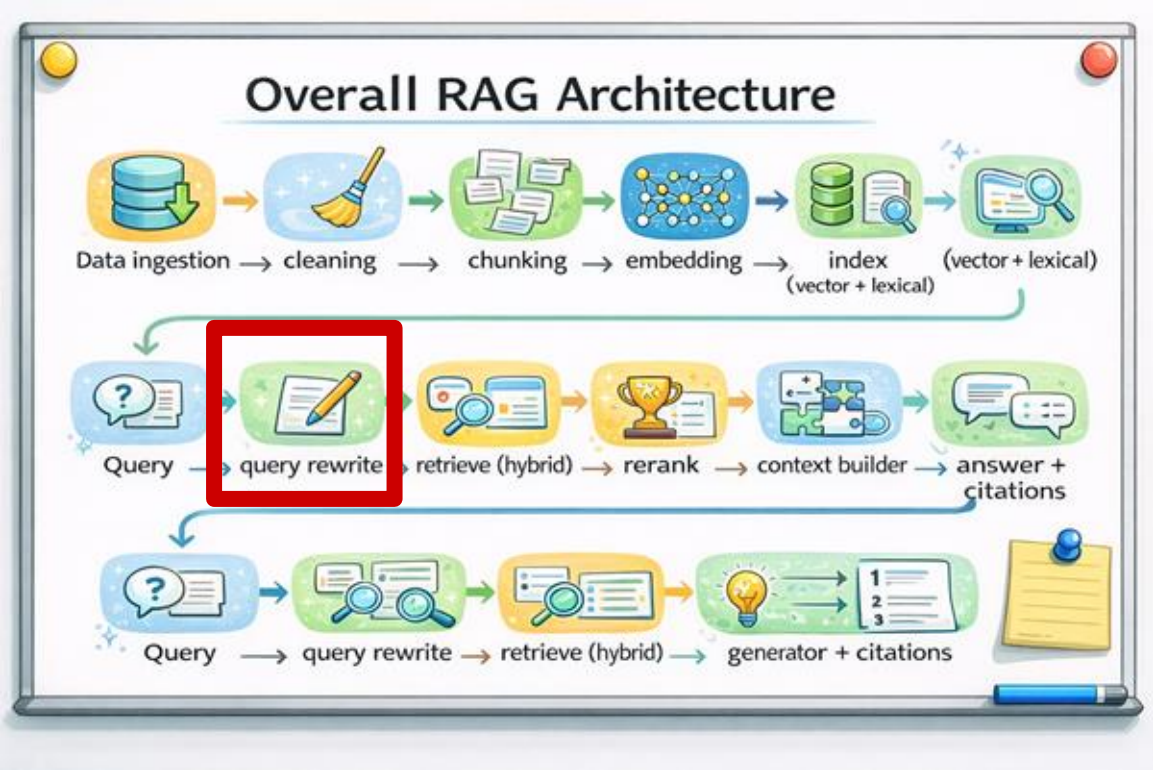

Overall RAG Architecture
Data ingestion
cleaning
chunking
embedding
index
(vector + lexical)
(vector + lexical)
Query
query rewrite
retrieve (hybrid)
rerank
context builder
answer + citations
Query
query rewrite
retrieve (hybrid)
generator + citations

We all have obviously typed like a maniac….

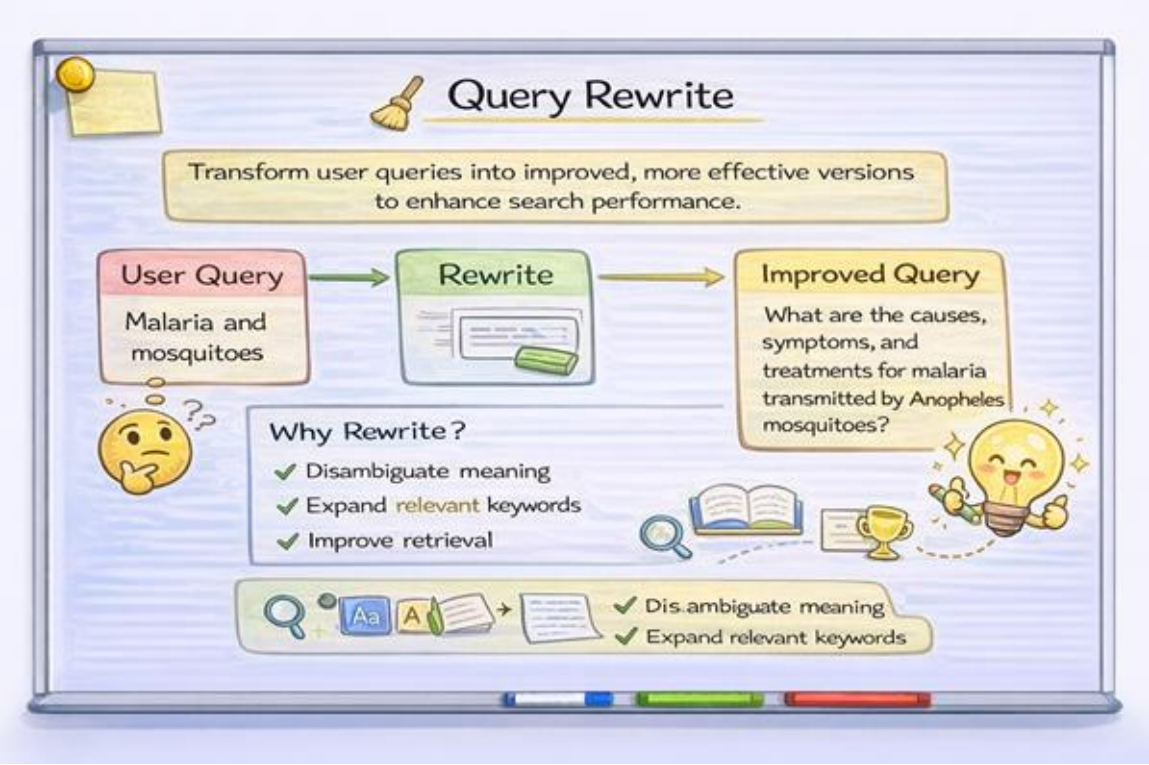


whst id a queru

RAG Architecture



Rewrite user query



Beyond term matching

- Query and document expansion
- Phrase search
- Term dependence
- Different document fields (e.g., title, body)
- Link analysis (e.g., PageRank)
- Learning to rank

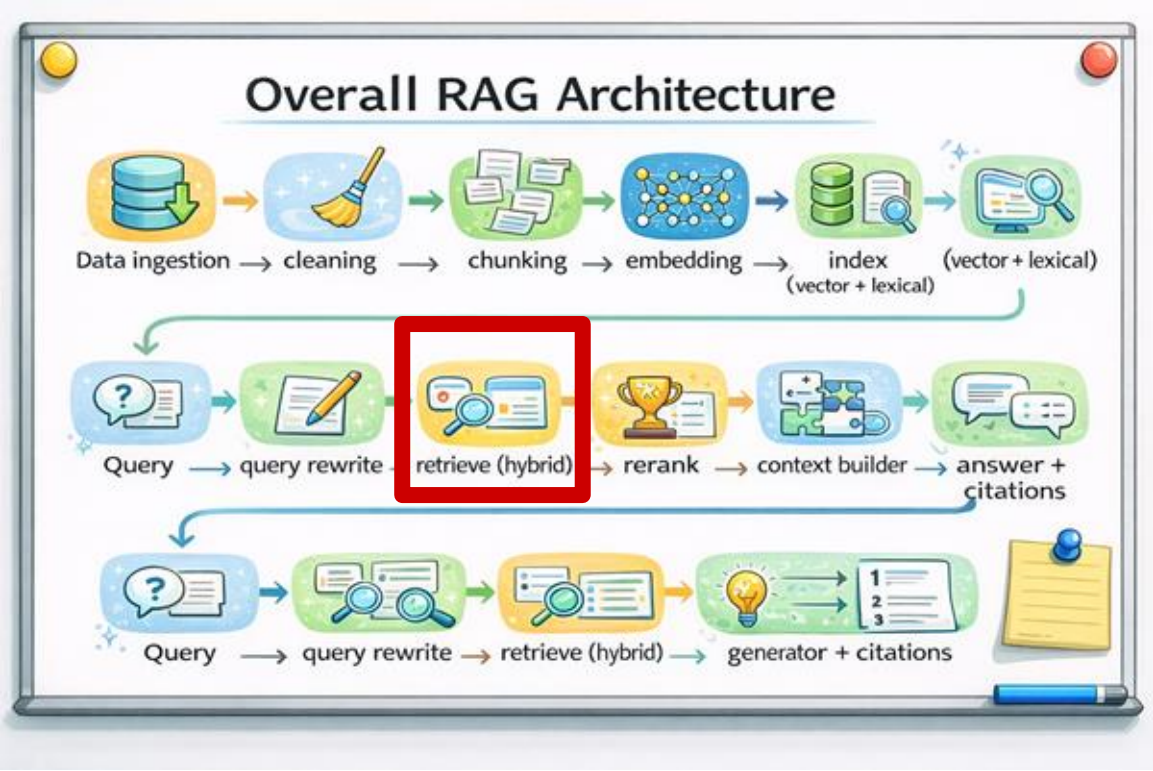




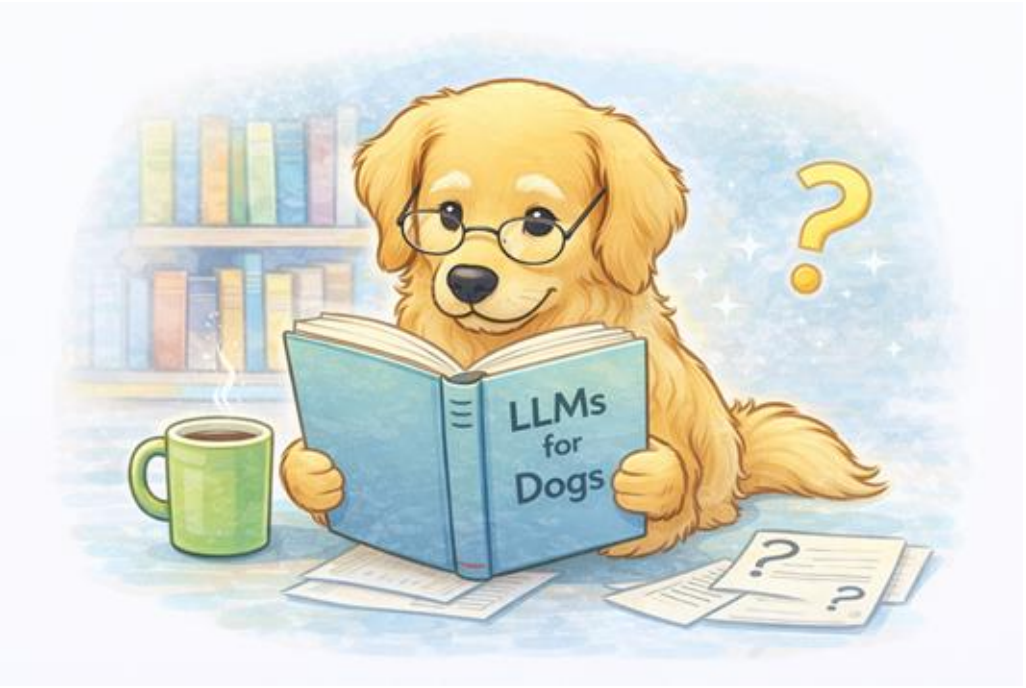




RAG Architecture

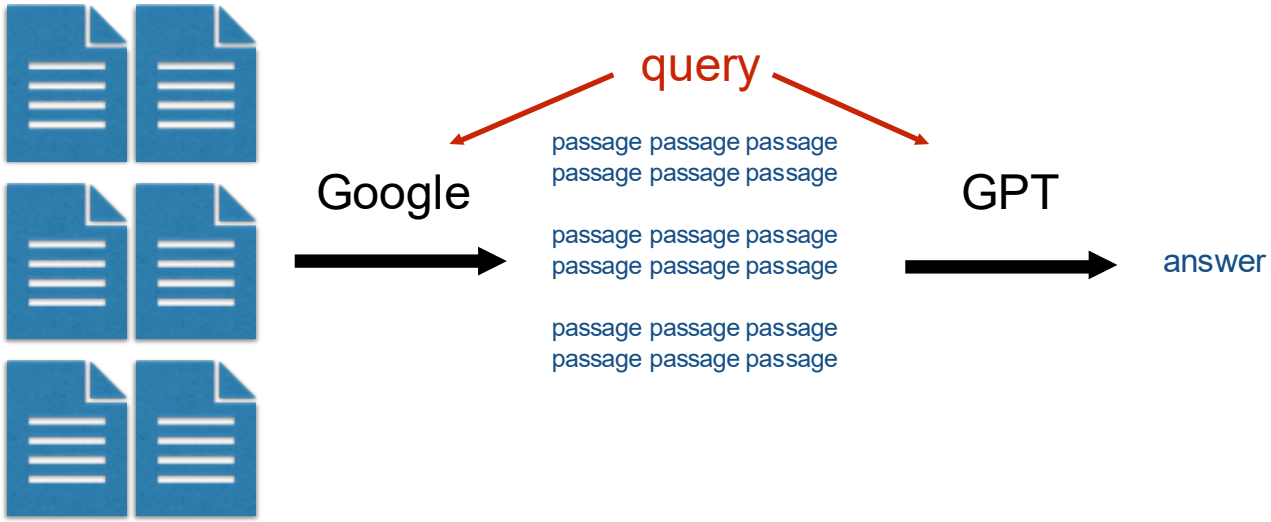


Retriever-Reader Models



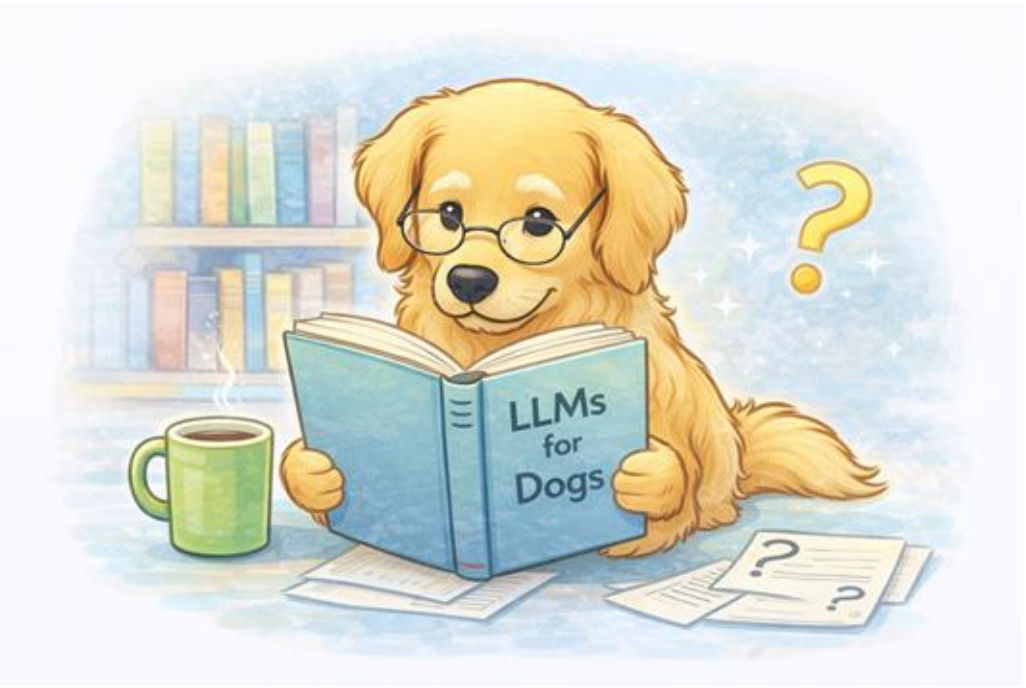
Simple: Just Chain Retrieval+Reading

- Use an out-of-the-box retriever and out-of-the-box reader

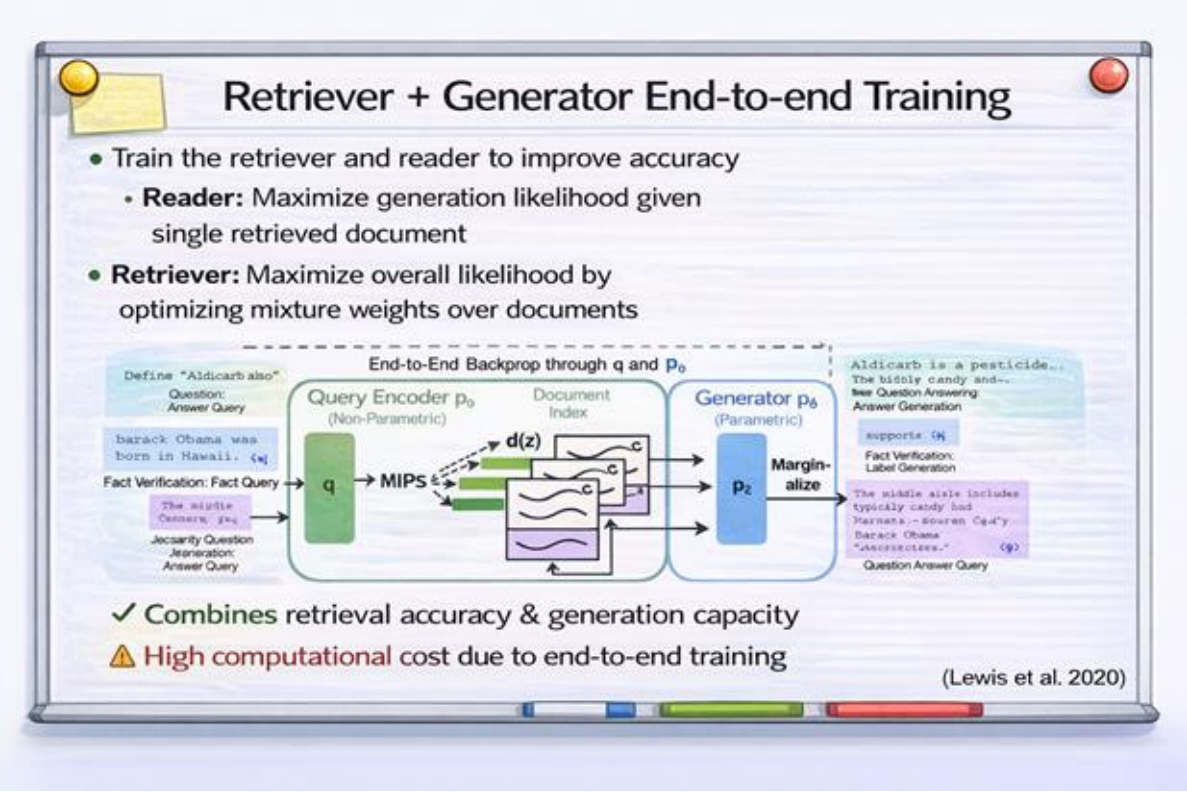


- Passages are concatenated to the context

The “Golden” Retriever-Reader Model



Retriever + Generator End-to-end Training



End-to-end Training Equations

End-to-End Training in RAG

- Generation combines retrieval + language modeling



Retriever selects doc z

Generator produces answer y

$$P_{RAG}(y|x) = \sum_z P_\eta(z|x) \cdot P_\theta(y|x,z)$$

✓ Retriever picks relevant documents

✓ Generator writes the answer using document

Retriever probability × Generator probability

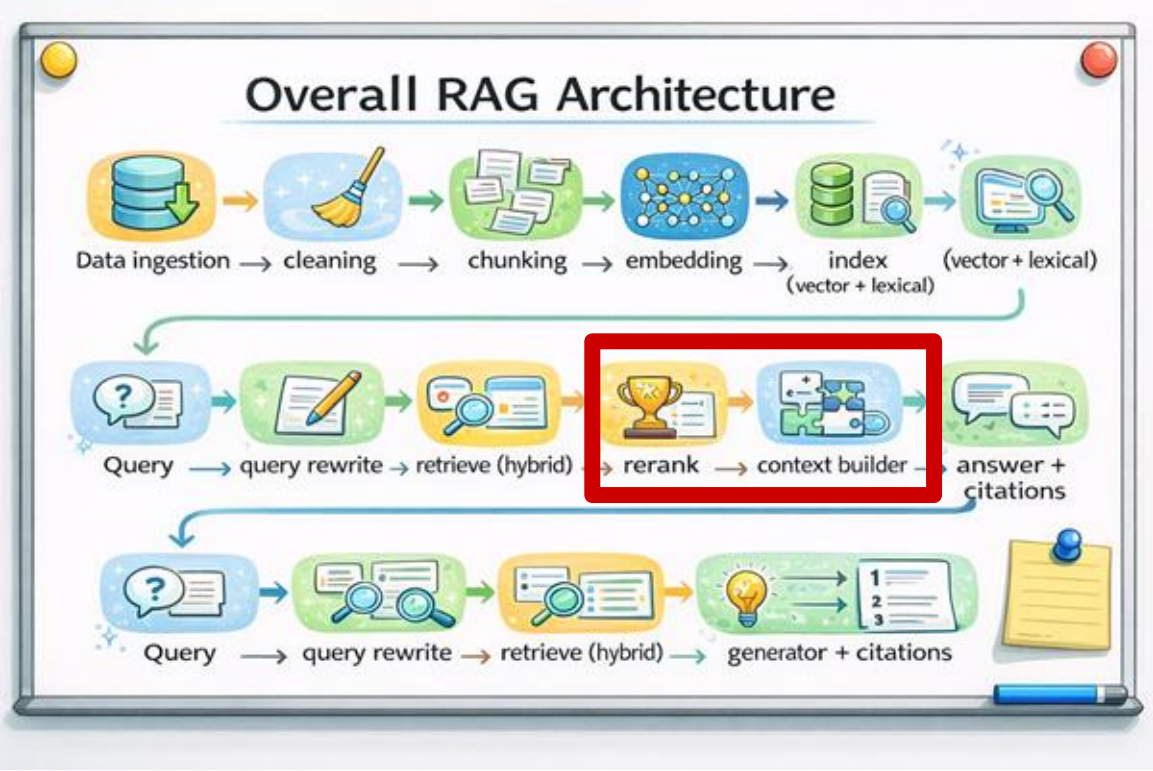
- Model marginalizes over possible documents
  - Considers multiple retrieved documents

- Model marginalizes over possible documents
  - Considers multiple retrieved documents

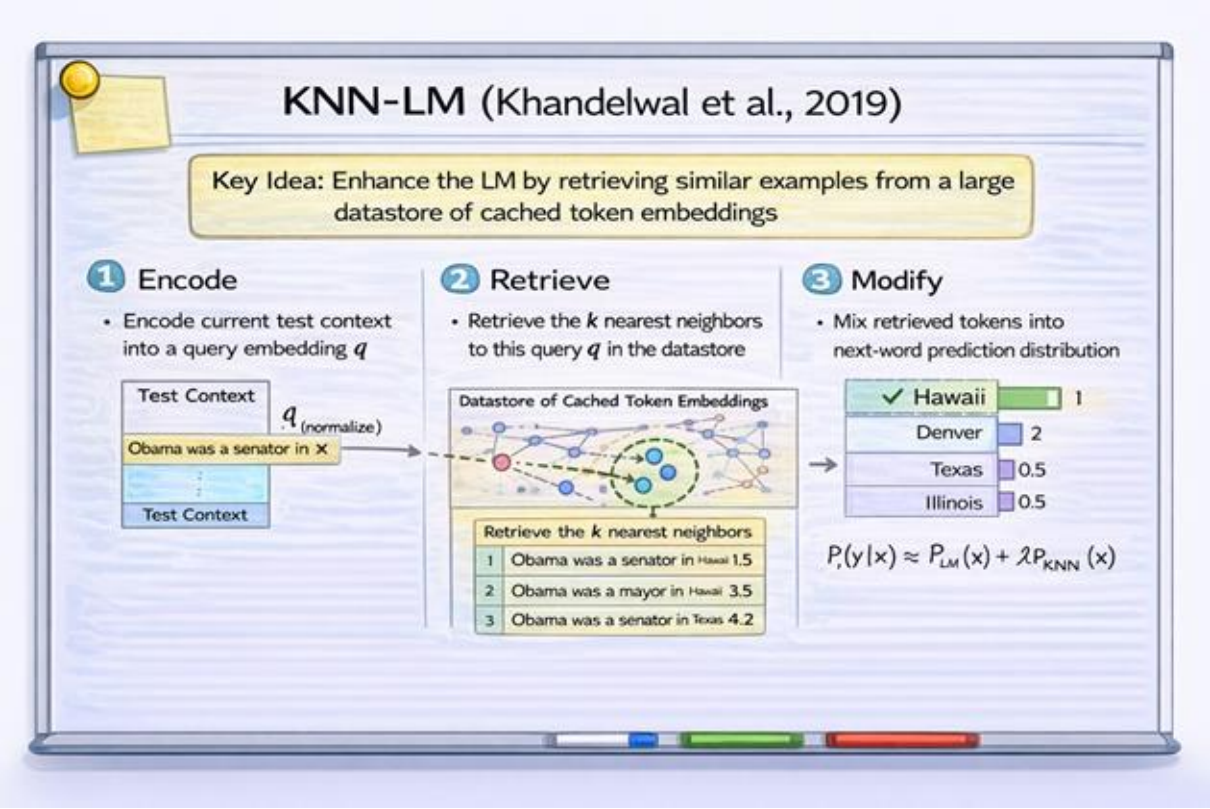
When do we Retrieve?

- **Once, at the beginning of generation**
  - Default method used by most systems (Lewis et al. 2020)
- **Several times during generation, as necessary**
  - Generate a search token (Schick et al. 2023)
  - Search when the model is uncertain (Jiang et al. 2023)
- **Every token**
  - Find similar final embeddings (Khandelwal et al. 2019)
  - Approximate attention with nearest neighbors (Bertsch et al. 2023)

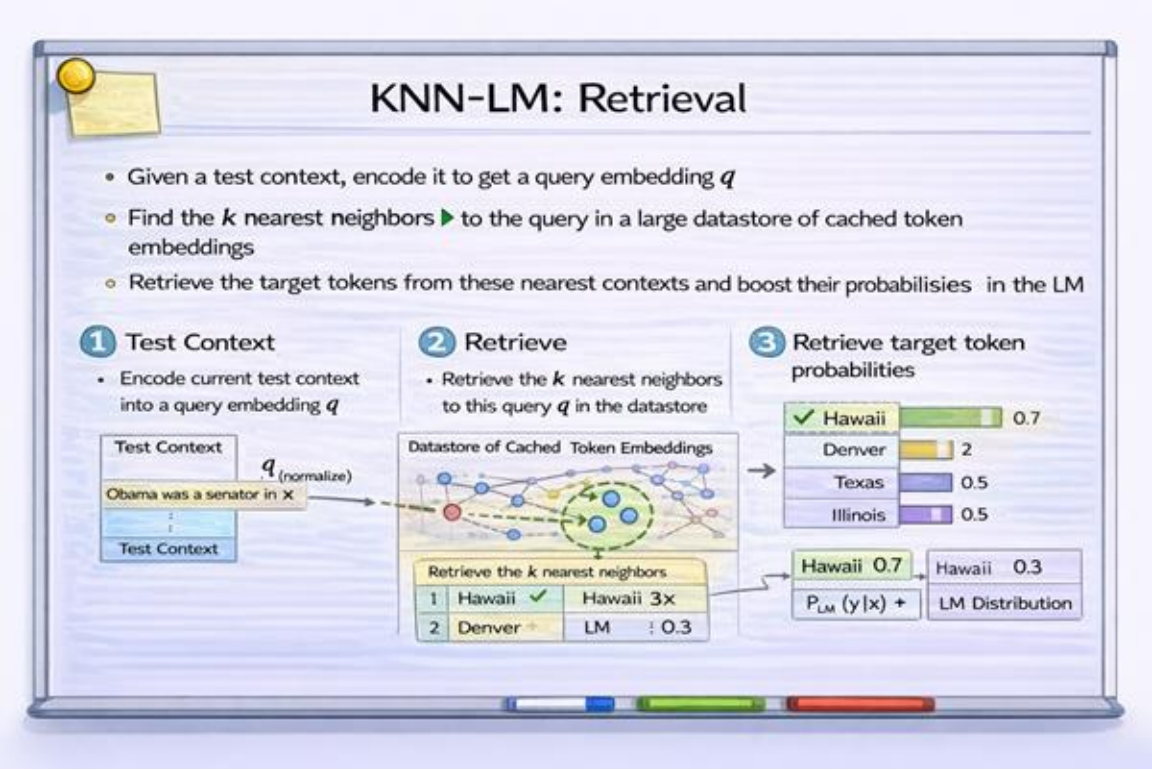
RAG Architecture



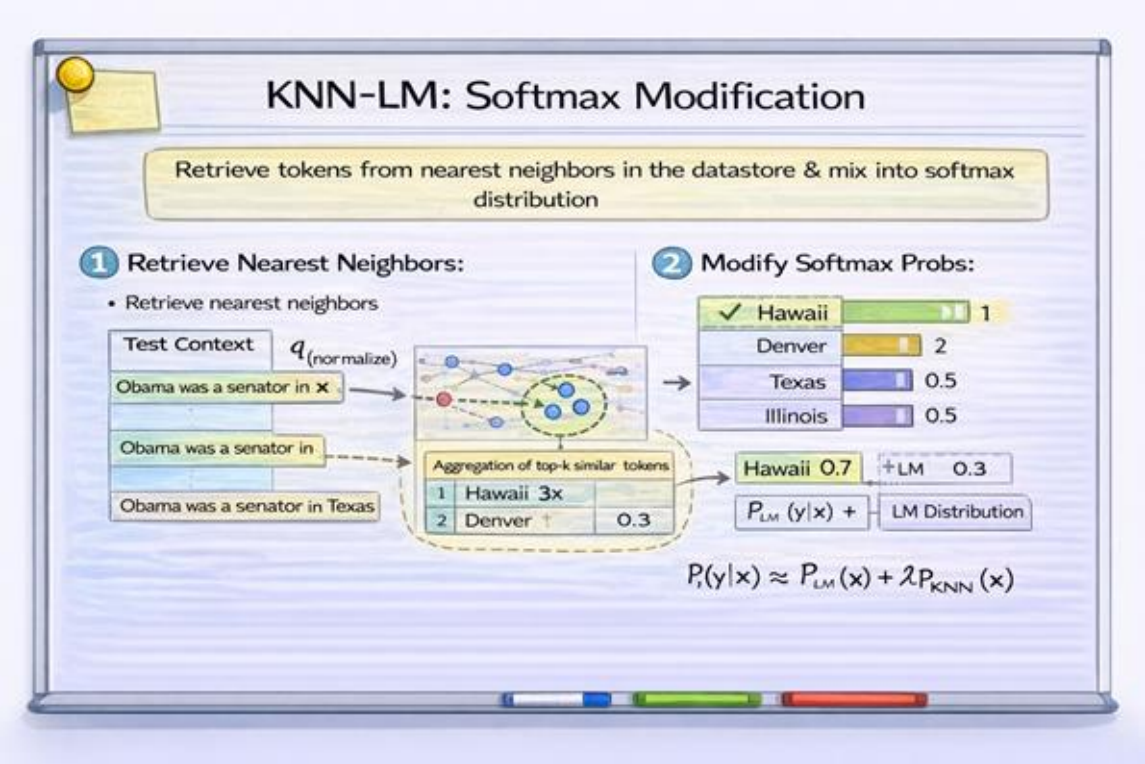
KNN-LM: An overview



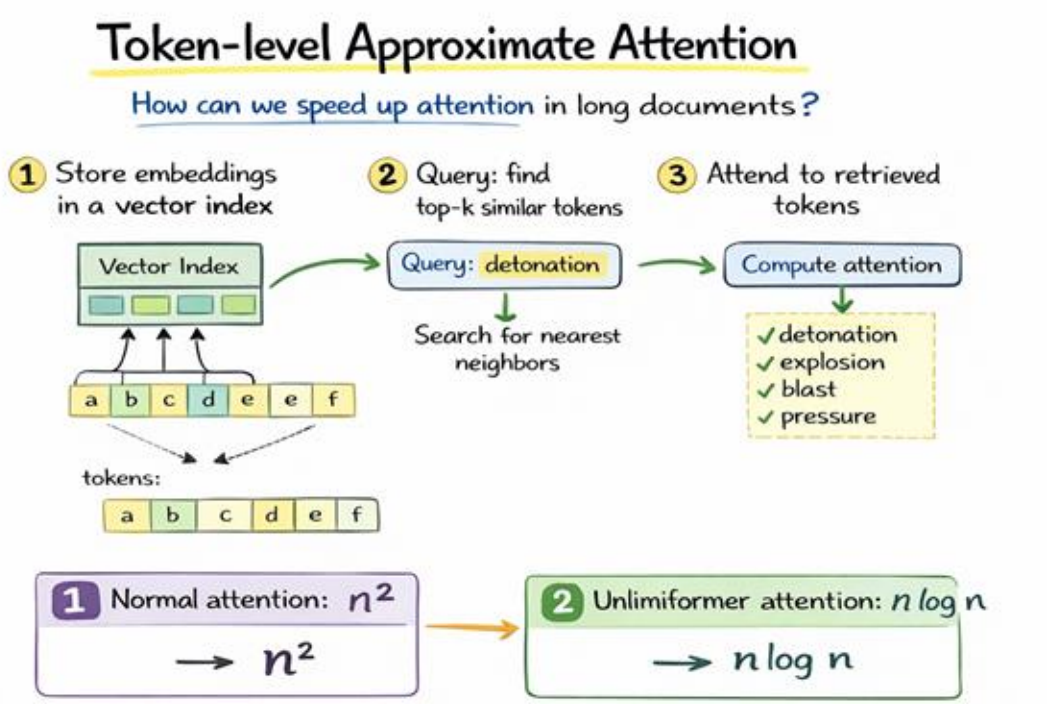
KNN-LM: Retrival



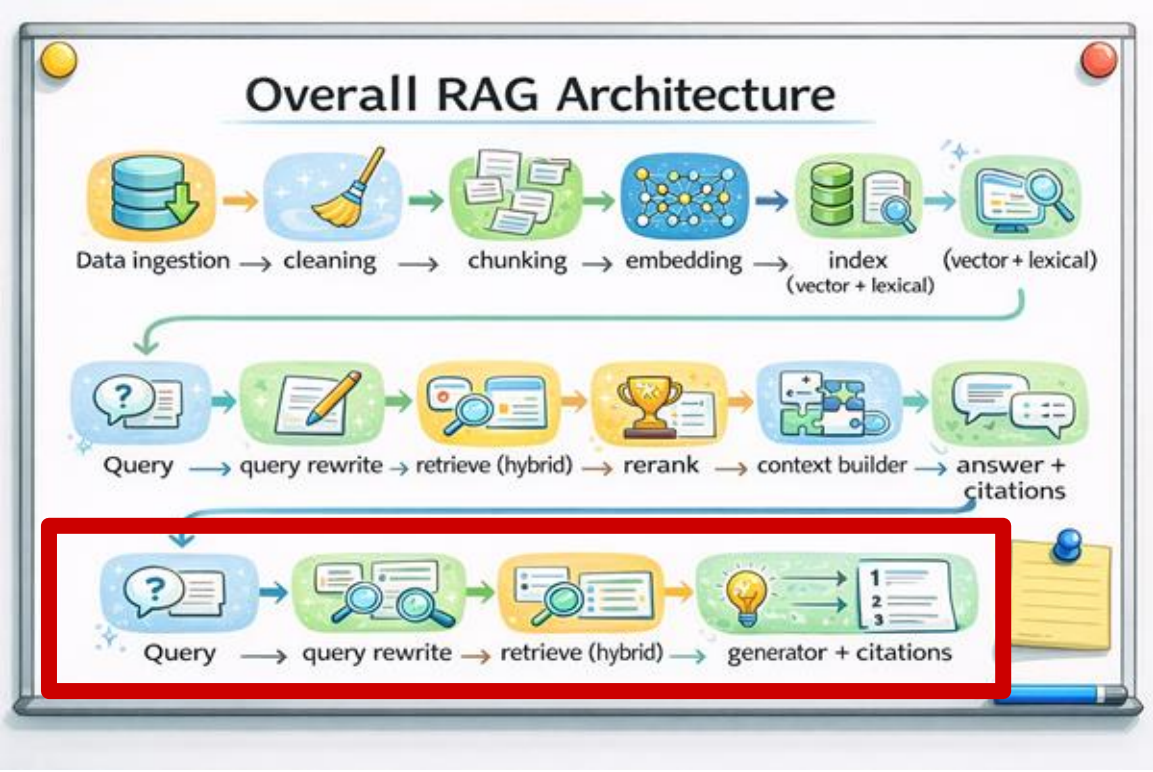
KNN-LM: Softmax Modification



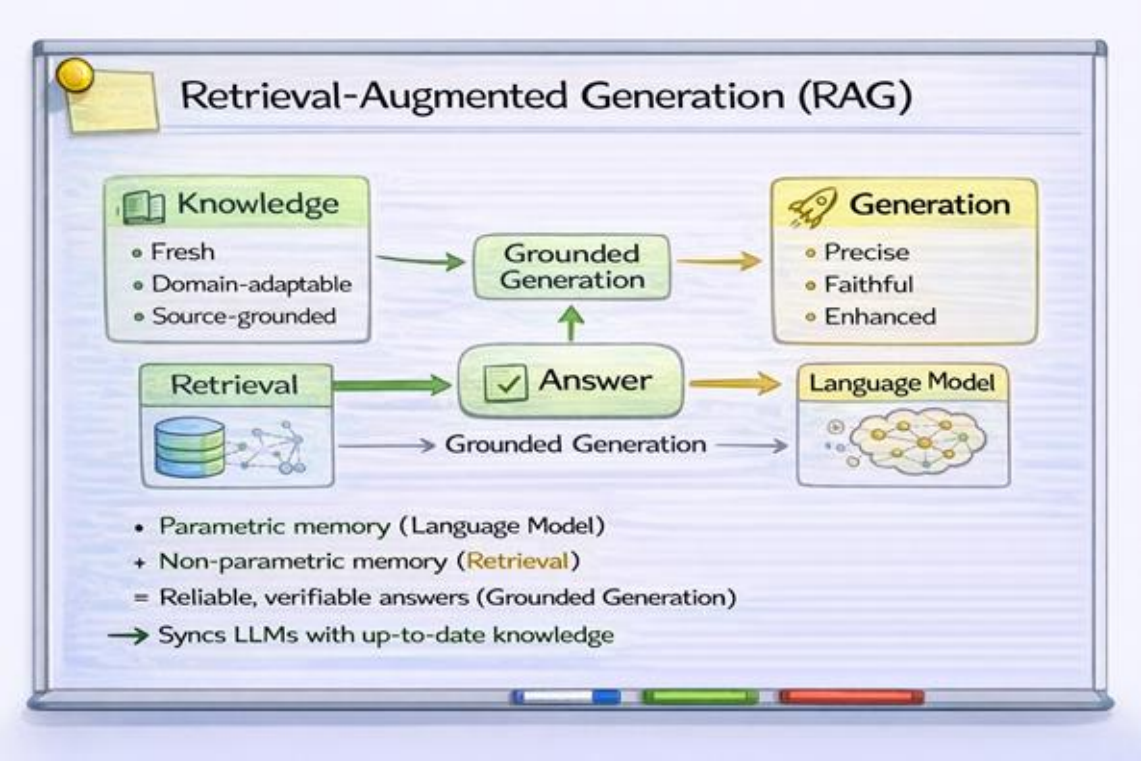
Token-level Approximate Attention



Finished RAG Pipeline



Summary


Retrieval-Augmented Generation (RAG)
Knowledge
Fresh
Domain-adaptable
Source-grounded
Grounded Generation
Generation
Precise
Faithful
Enhanced
Retrieval
Answer
Language Model
Grounded Generation
Parametric memory (Language Model)
+ Non-parametric memory (Retrieval)
= Reliable, verifiable answers (Grounded Generation)
Syncs LLMs with up-to-date knowledge