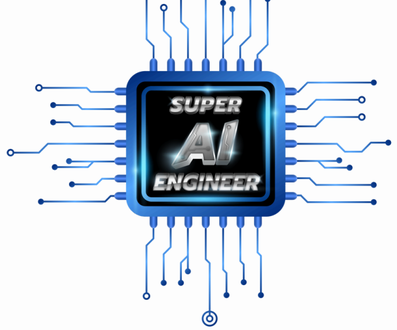


AiAT
ARTIFICIAL INTELLIGENCE ASSOCIATION OF THAILAND

LLM-POWERED RAG: BUILDING AI FROM STRUCTURED DATABASES AND UNSTRUCTURED PDFS

Presented by:

Konthee Boonmeeprakob

21 Apr 2025

About Us

Name : Konthee Boonmeeprakob (Knot)

Education :

- BSE & MSE in Physics, Prince of Songkla University

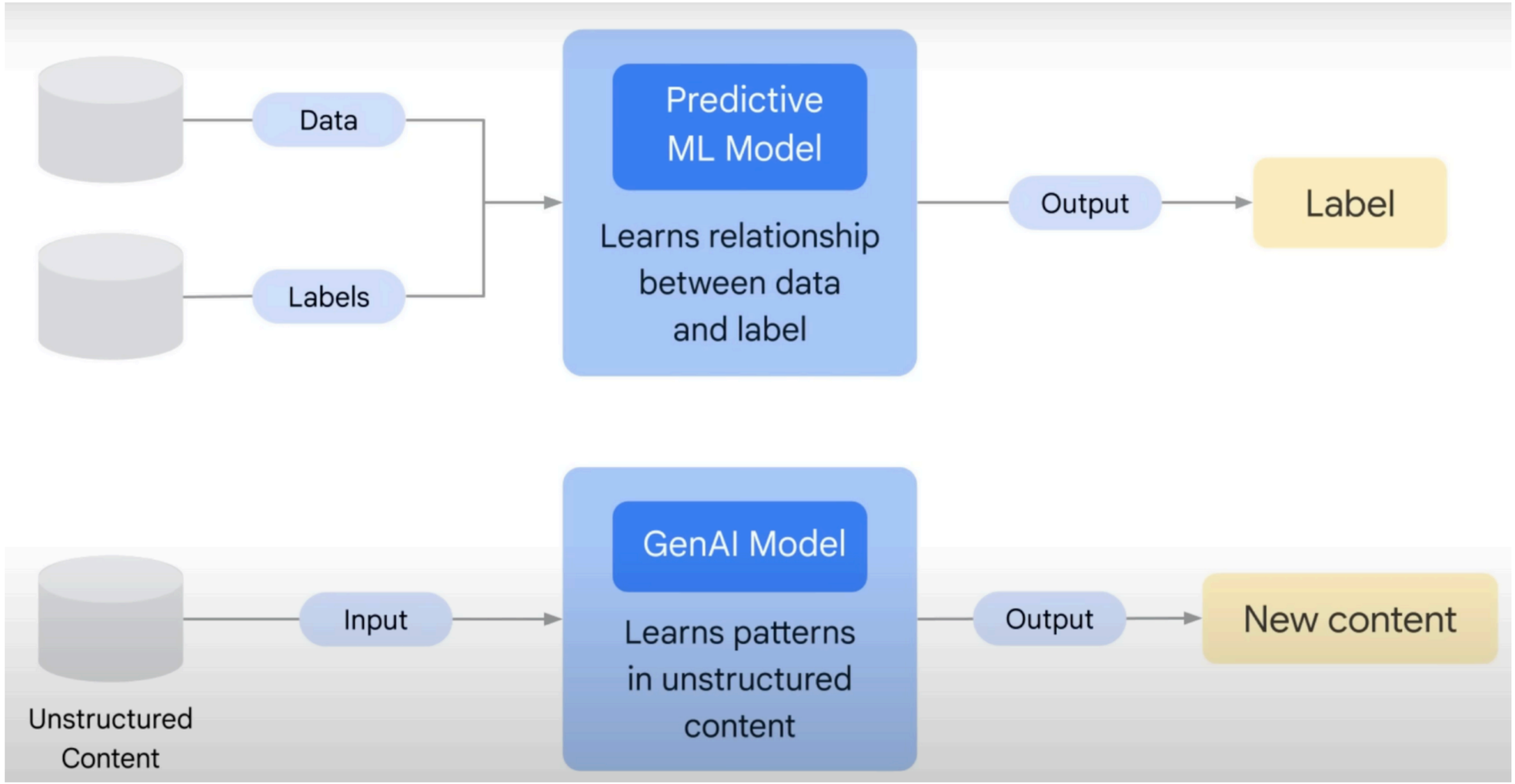
Work:

- Super Ai Engineer SS3
- Data Scientist at BDI

Contact :

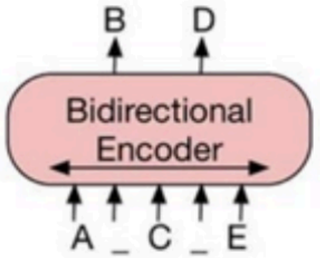
- Konthee1995@gmail.com

Generative AI, different from AI

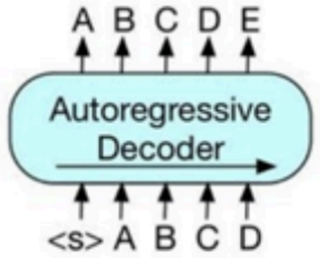


Type of Transformer

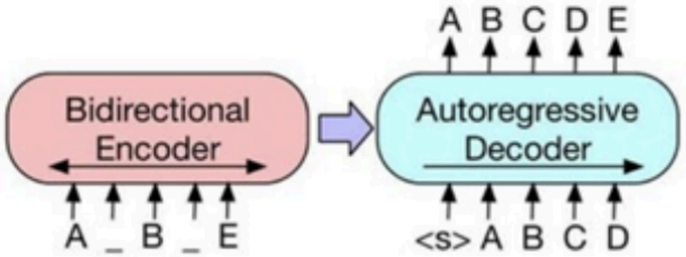
Differences of BERT, GPT, and BART (Lewis et al., 2019)



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



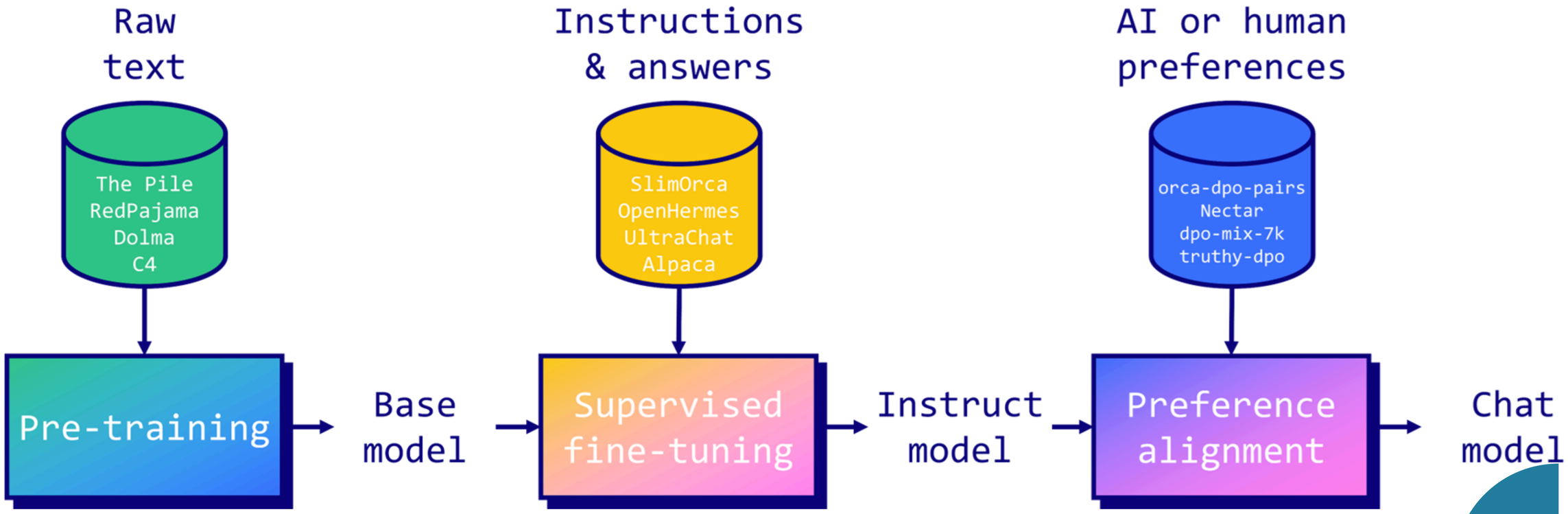
(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with a mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

- **BERT:**
  - Bidirectional encoder
- **GPT:**
  - Generative Pretrained Transformer
  - Autoregressive (unidirectional) decoder
- **BART:**
  - Bidirectional encoder + autoregressive decoder

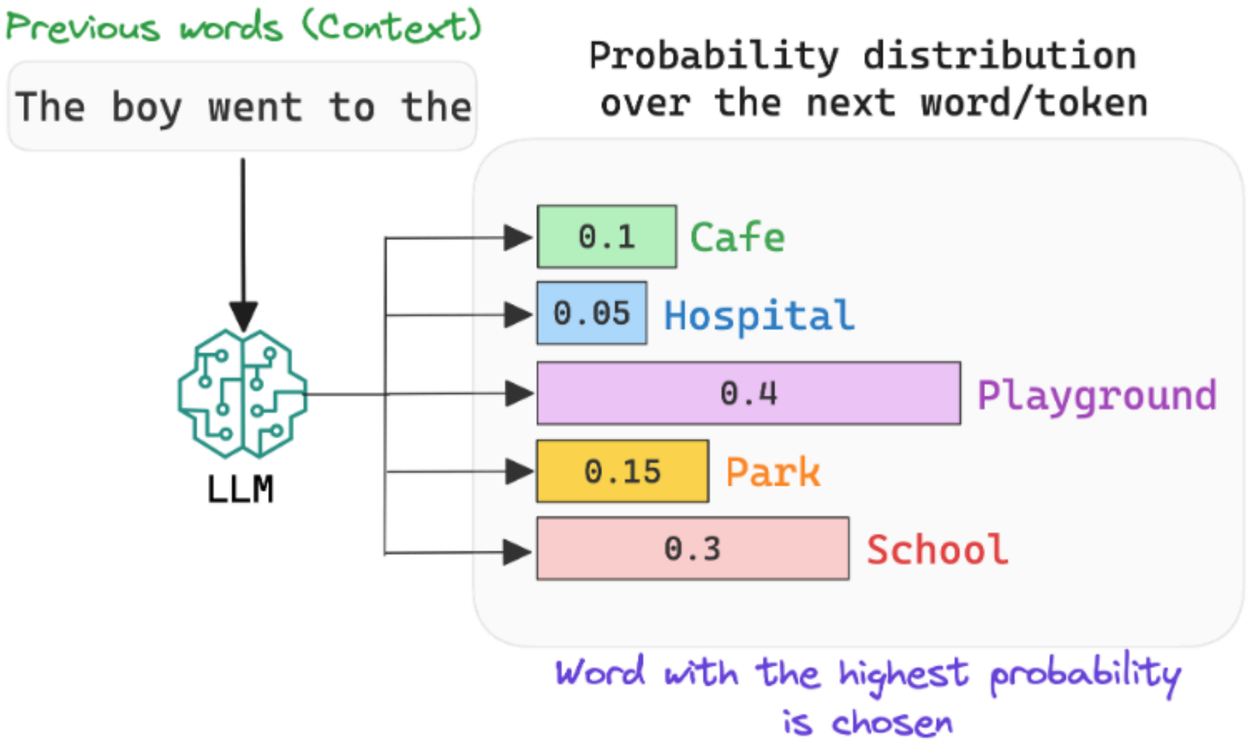
How to Build a LLM



https://huggingface.co/blog/mlabonne/sft-llama3

Pre-train

Next word prediction!



Pre-train

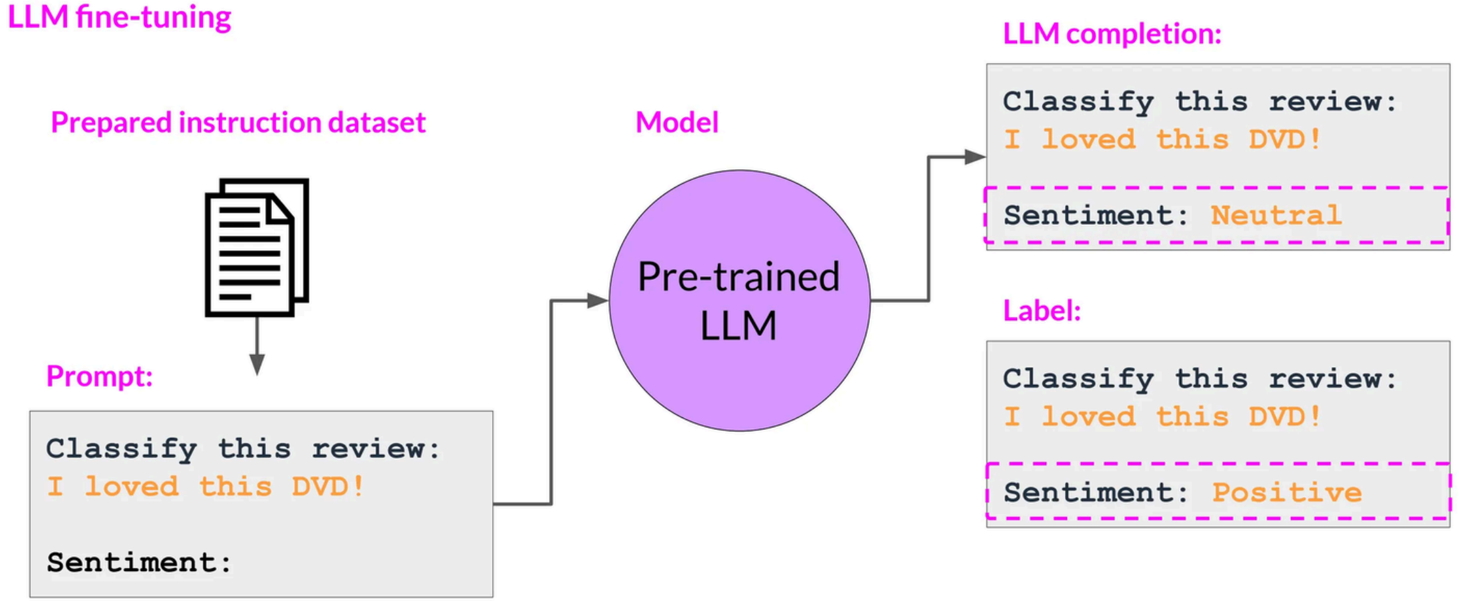
Loss calculation ⁉

Loss = -log(P('Playground'/'The boy went to')) = -log(0.4)

Cross-entropy loss / Negative log-likelihood

**Llama 3.1 has been pretrained on over 15 trillion tokens

Supervised Fine-Tuning (SFT)



Supervised Fine-Tuning (SFT)

Instruction Dataset (Ouyang et al., 2022)

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

Use Case	Example
brainstorming	List five ideas for how to regain enthusiasm for my career
brainstorming	What are some key points I should know when studying Ancient Greece?
brainstorming	What are 4 questions a user might have after reading the instruction manual for a trash compactor? {user manual} 1.

- Prompt: “instruction”
  - Cleverly designed set of instructions and responses for a chatbot
  - Covering frequently asked questions and their answers

(Ouyang et al., 2022)

Supervised Fine-Tuning (SFT)

closed qa	Answer the following question: What shape is the earth? A) A circle B) A sphere C) An ellipse D) A plane
closed qa	Tell me how hydrogen and helium are different, using the following facts: {list of facts}
open qa	I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown". Q: What is human life expectancy in the United States? A: Human life expectancy in the United States is 78 years. Q: Who was president of the United States in 1955? A:
open qa	Who built the statue of liberty?
open qa	How do you take the derivative of the sin function?
open qa	who are the indiginous people of New Zealand?

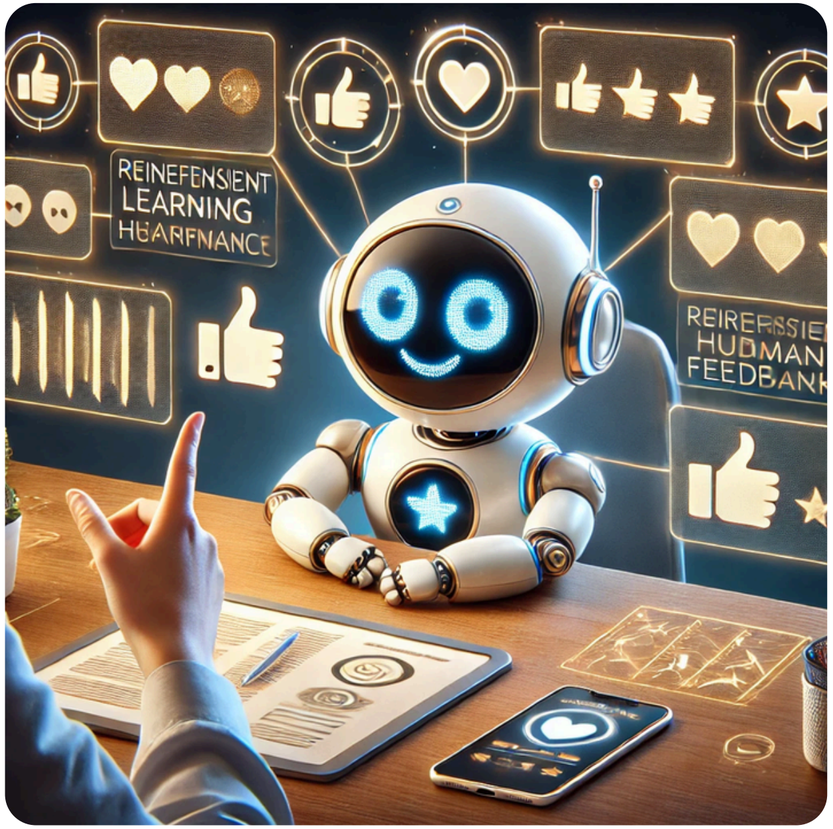
(Ouyang et al., 2022)

Supervised Fine-Tuning (SFT)

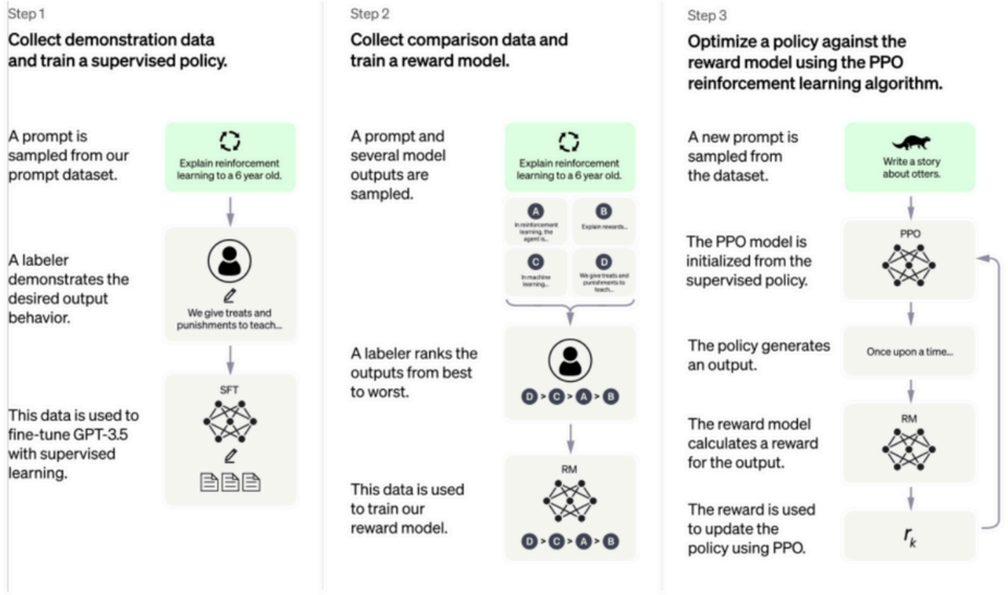
Use Case	Example
classification	The following is a list of companies and the categories they fall into: Apple, Facebook, Fedex Apple Category: Technology Facebook Category: Social Media Fedex Category:
extract	Text: {text} Keywords:
generation	"Hey, what are you doing there?" Casey was startled. He hadn't even begun to
generation	The name of the next Star Wars movie is
generation	This is the research for an essay: === {description of research} === Write a high school essay on these topics: ===

(Ouyang et al., 2022)

Reinforcement learning from human feedback (RLHF)



Prompts + Human Ranking + RL



- 3 steps
  - Fine-tune the language model with the instruction dataset
  - Retrain the reward model for chat response with human ranking
  - Optimize the policy model w.r.t. the reward model with the PPO Algorithm (proximal policy optimization)

(Ouyang et al., 2022)

How to Build a LLM

1) PRE-TRAINING

2) SUPERVISED FINETUNING

3) REWARD-FINETUINE

Type of LLM

Base LLM

Predicts next word, based on text training data

Once upon a time, there was a unicorn
that lived in a magical forest with
all her unicorn friends

What is the capital of France?
What is France's largest city?
What is France's population?
What is the currency of France?

Instruction Tuned LLM

Tries to follow instructions

Fine-tune on instructions and
good attempts at following those instructions.

RLHF: Reinforcement Learning with
Human Feedback

Helpful, Honest, Harmless

What is the capital of France?
The capital of France is Paris.

Pitfalls of LLMs

LLMs are extremely powerful, but they are by no means perfect.
There are many pitfalls that you should be aware of when using them.

Bias

They will sometimes say sexist/racist/homophobic things. Be careful when using LLMs in consumer-facing applications.

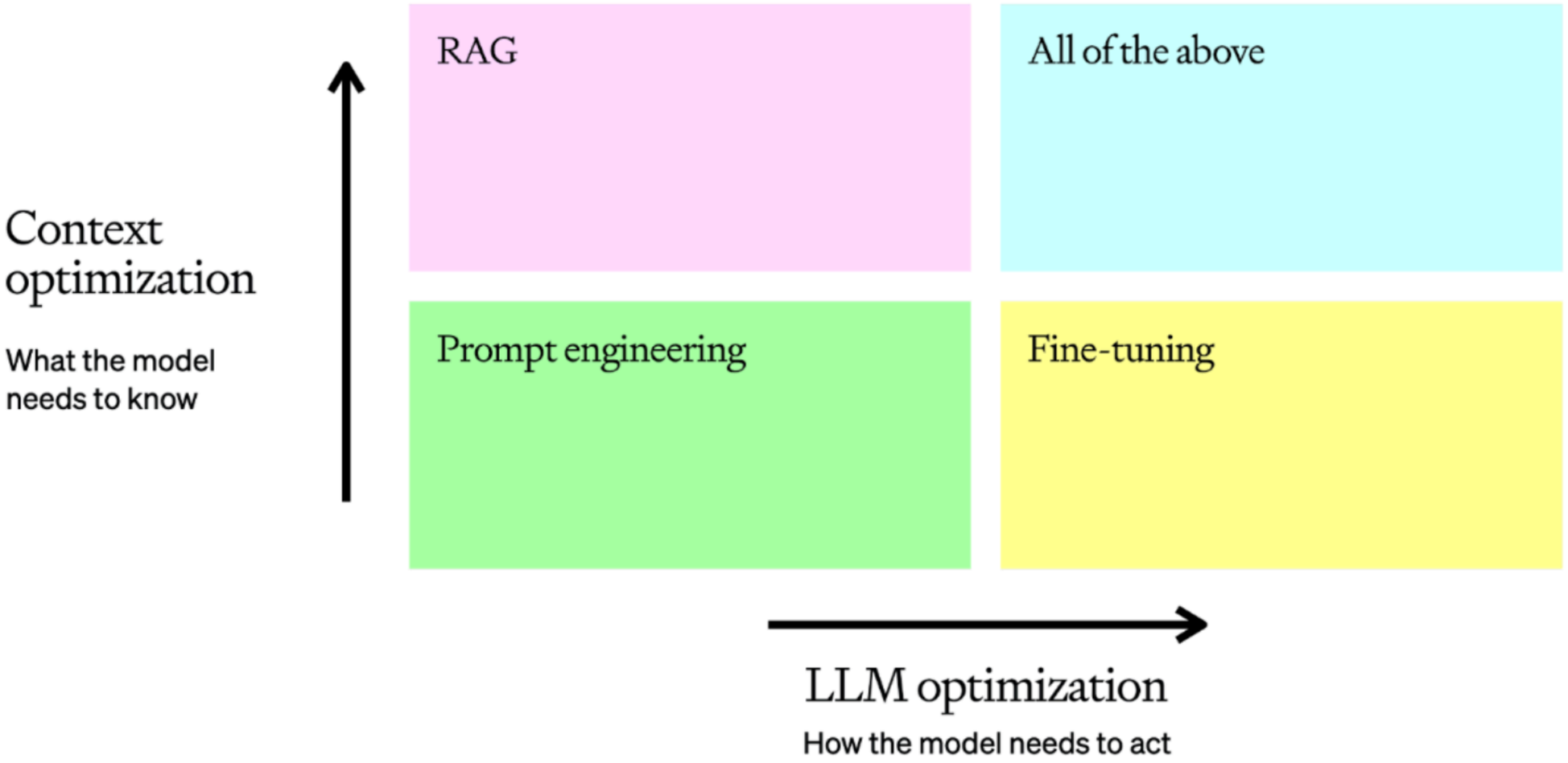
Hallucinations

LLMs will frequently generate falsehoods when asked a question that they do not know the answer to.

Math

LLMs are often bad at math. They are often unable to solve complex math problems.

LLM optimization context

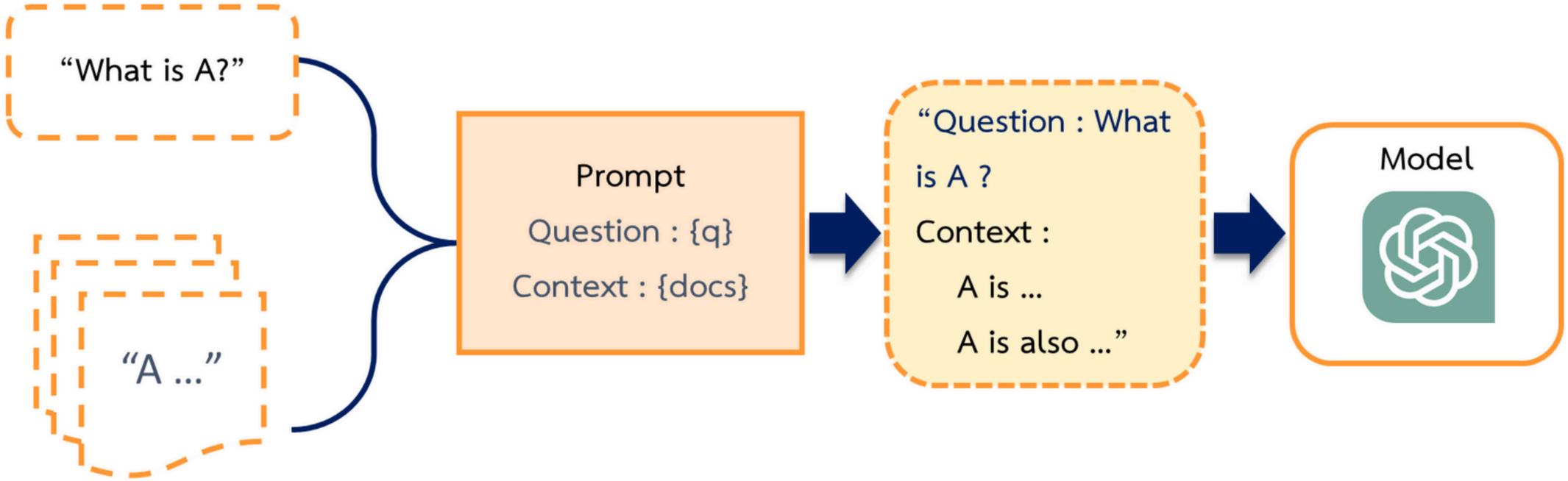


RAG

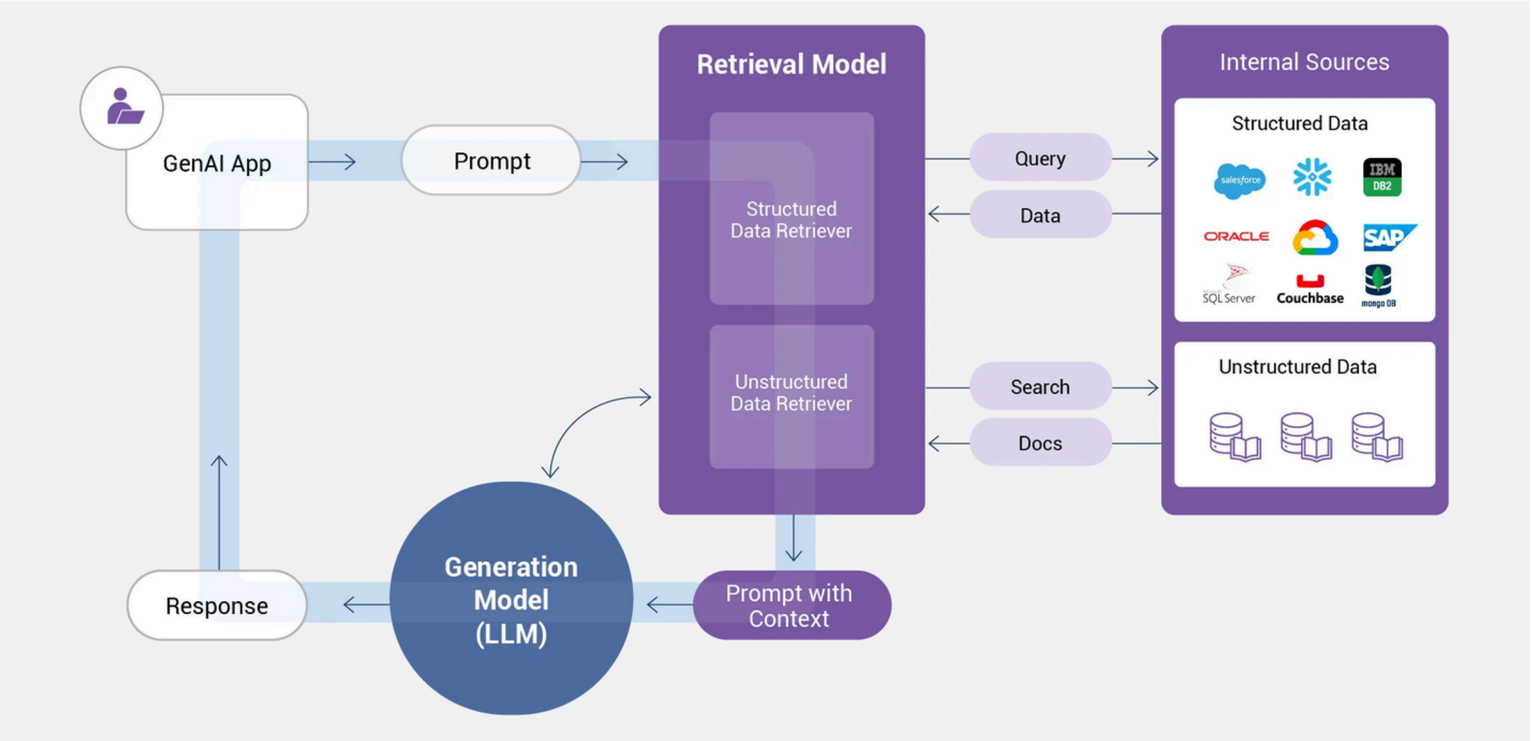
RETRIEVAL-AUGMENTED GENERATION

RAG

Retrieval-Augmented Generation



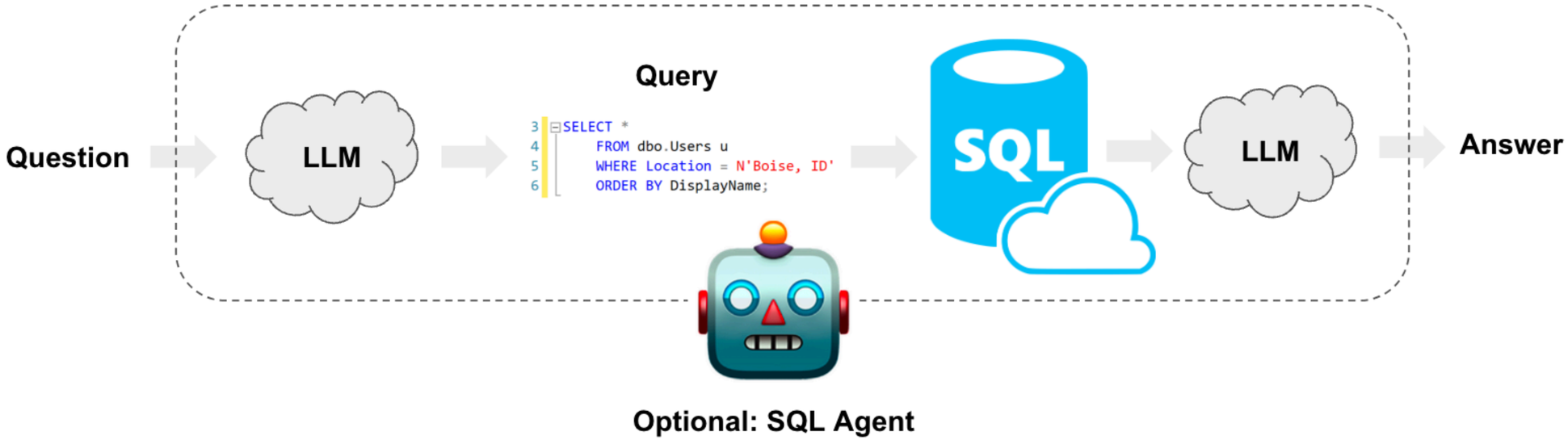
RAG



- Structured data: using prompt design to create SQL code for querying result from a database
- Unstructured data: using embedding model to transform text and the resulting vector through similarity search with a vector database

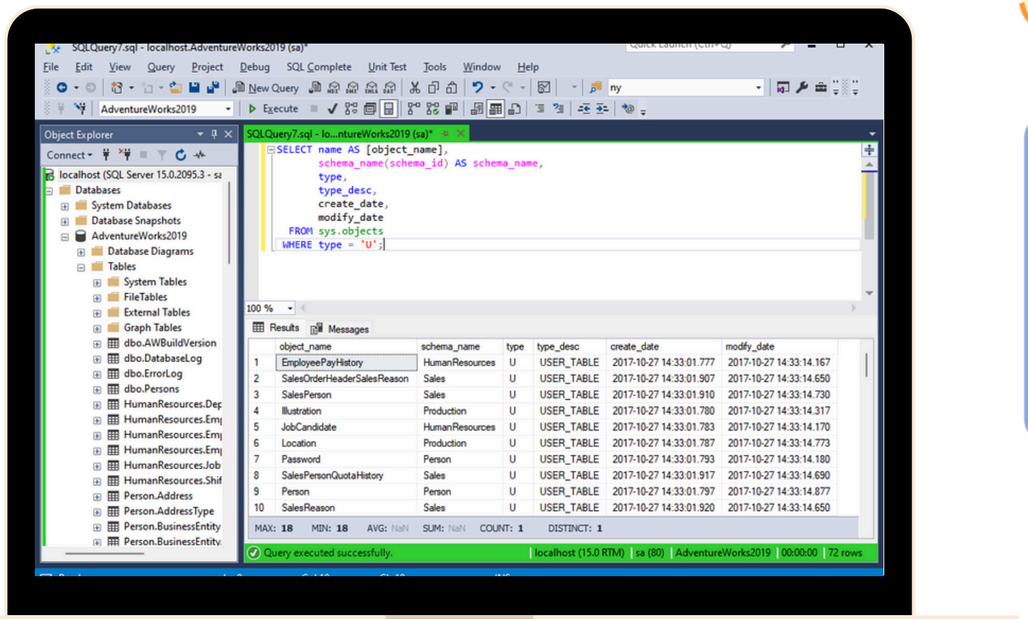
Structured RAG

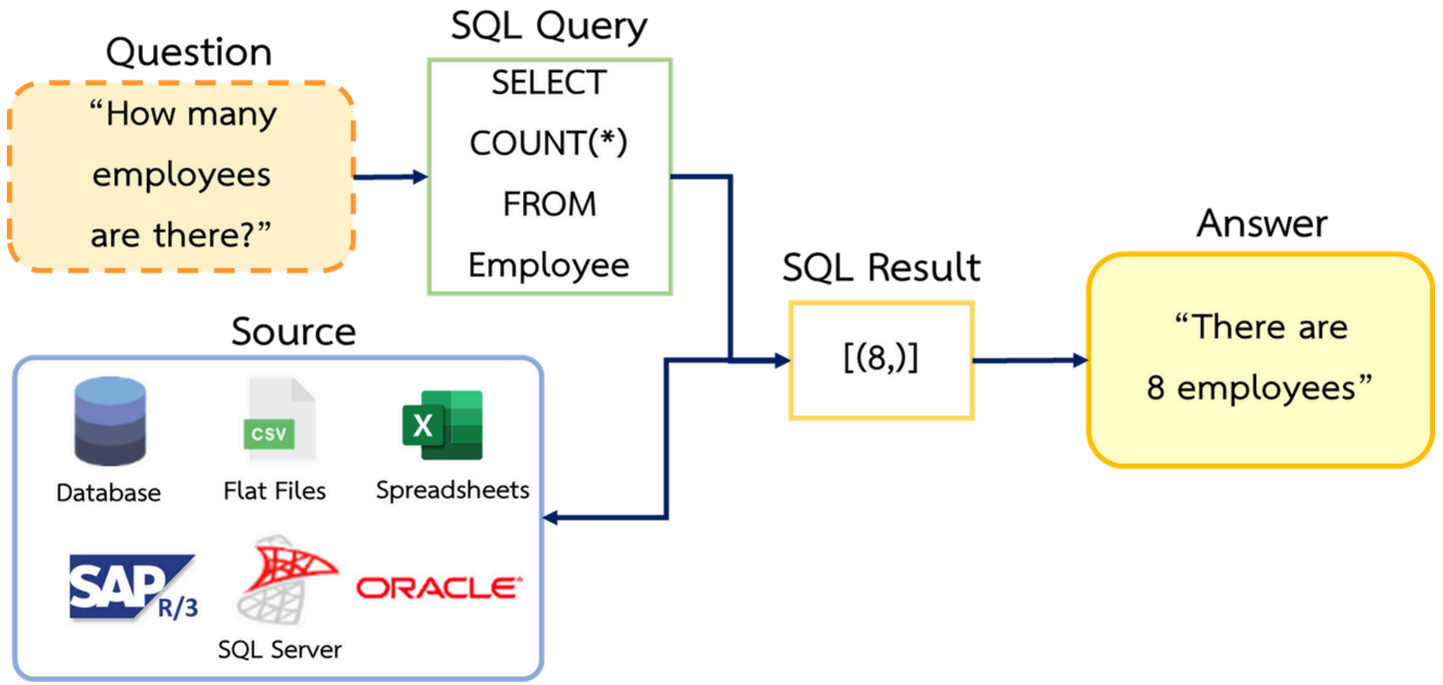
Question/Answering system over SQL data



https://python.langchain.com/docs/tutorials/sql_qa/

Structured RAG





Sql dataset

Split (1)
train · 78.6k rows

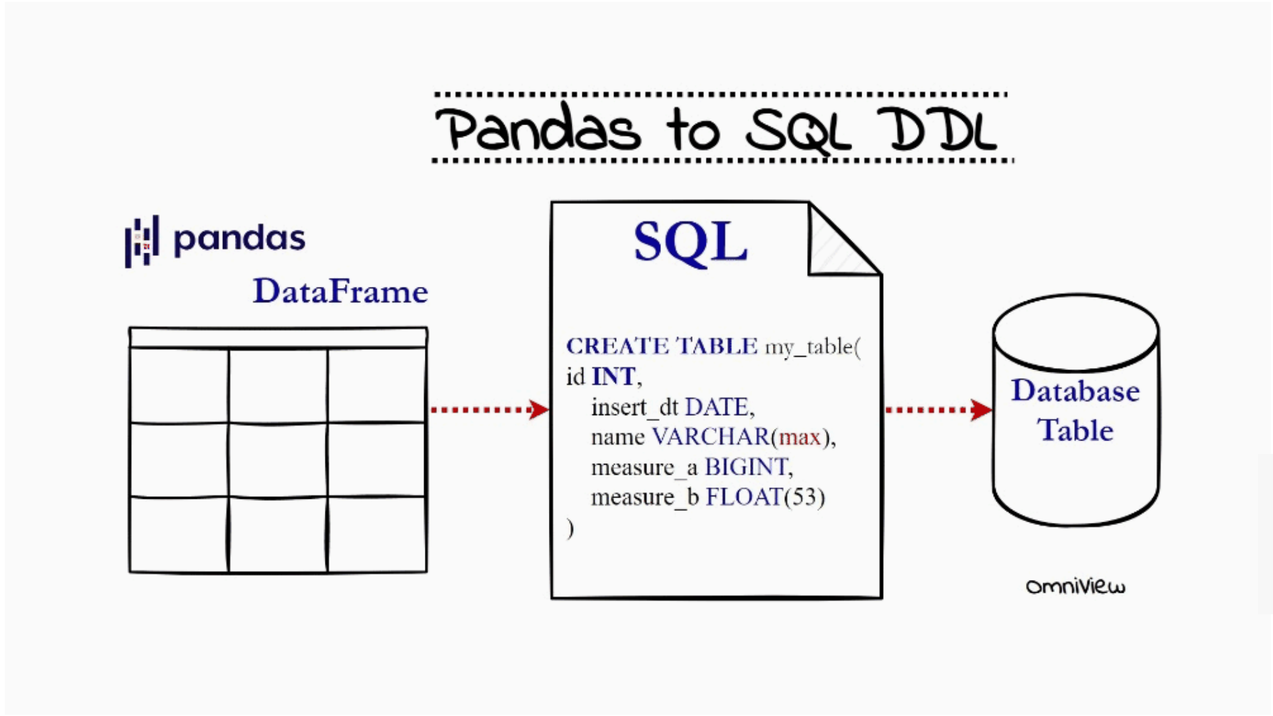
Search this dataset

SQL Console

answer string · lengths 18 557	question string · lengths 12 244	context string · lengths 27 489
SELECT COUNT(*) FROM head WHERE age > 56	How many heads of the departments are older than 56 ?	CREATE TABLE head (age INTEGER)
SELECT name, born_state, age FROM head ORDER BY age	List the name, born state and age of the heads of departments ordered by age.	CREATE TABLE head (name VARCHAR, born_state VARCHAR, age VARCHAR)
SELECT creation, name, budget_in_billions FROM department	List the creation year, name and budget of each department.	CREATE TABLE department (creation VARCHAR, name VARCHAR, budget_in_billions VARCHAR)
SELECT MAX(budget_in_billions), MIN(budget_in_billions) FROM department	What are the maximum and minimum budget of the departments?	CREATE TABLE department (budget_in_billions INTEGER)
SELECT AVG(num_employees) FROM department WHERE ranking BETWEEN 10 AND 15	What is the average number of employees of the departments whose rank is between 10 and 15?	CREATE TABLE department (num_employees INTEGER, ranking INTEGER)
SELECT name FROM head WHERE born_state <> 'California'	What are the names of the heads who are born outside the California state?	CREATE TABLE head (name VARCHAR, born_state VARCHAR)
SELECT DISTINCT T1.creation FROM department AS T1 JOIN management AS T2 ON T1.department_id = T2.department_id JOIN head AS T3 ON T2.head_id = T3.head_id…	What are the distinct creation years of the departments managed by a secretary born in state 'Alabama'?	CREATE TABLE department (creation VARCHAR, department_id VARCHAR); CREATE TABLE management (department_id VARCHAR, head_id VARCHAR); CREATE TABLE head (head_i…
SELECT born_state FROM head GROUP BY born_state HAVING COUNT(*) >= 3	What are the names of the states where at least 3 heads were born?	CREATE TABLE head (born_state VARCHAR)
SELECT creation FROM department GROUP BY creation ORDER BY COUNT(*) DESC LIMIT 1	In which year were most departments established?	CREATE TABLE department (creation VARCHAR)
SELECT T1.name, T1.num_employees FROM department AS T1 JOIN management AS T2 ON T1.department_id = T2.department_id WHERE T2.temporary_acting = 'Yes'	Show the name and number of employees for the departments managed by heads whose temporary acting value is 'Yes'?	CREATE TABLE management (department_id VARCHAR, temporary_acting VARCHAR); CREATE TABLE department (name VARCHAR, num_employees VARCHAR, department_id…
SELECT COUNT(DISTINCT temporary_acting) FROM management	How many acting statuses are there?	CREATE TABLE management (temporary_acting VARCHAR)
SELECT COUNT(*) FROM department WHERE NOT department_id IN (SELECT department_id FROM management)	How many departments are led by heads who are not mentioned?	CREATE TABLE management (department_id VARCHAR); CREATE TABLE department (department_id VARCHAR)
SELECT DISTINCT T1.age FROM management AS T2 JOIN head AS T1 ON T1.head_id = T2.head_id WHERE T2.temporary_acting = 'Yes'	What are the distinct ages of the heads who are acting?	CREATE TABLE head (age VARCHAR, head_id VARCHAR); CREATE TABLE management (head_id VARCHAR, temporary_acting VARCHAR)
SELECT T3.born_state FROM department AS T1 JOIN management AS T2 ON T1.department_id = T2.department_id JOIN head AS T3 ON T2.head_id = T3.head_id…	List the states where both the secretary of 'Treasury' department and the secretary of 'Homeland Security' were born.	CREATE TABLE management (department_id VARCHAR, head_id VARCHAR); CREATE TABLE head (born_state VARCHAR, head_id VARCHAR); CREATE TABLE department…
SELECT T1.department_id, T1.name, COUNT(*) FROM management AS T2 JOIN department AS T1 ON T1.department_id = T2.department_id GROUP BY…	Which department has more than 1 head at a time? List the id, name and the number of heads.	CREATE TABLE management (department_id VARCHAR); CREATE TABLE department (department_id VARCHAR, name VARCHAR)
SELECT head_id, name FROM head WHERE name LIKE '%Ha%'	Which head's name has the substring 'Ha'? List the id and name.	CREATE TABLE head (head_id VARCHAR, name VARCHAR)
SELECT COUNT(*) FROM farm	How many farms are there?	CREATE TABLE farm (Id VARCHAR)
SELECT Total_Horses FROM farm ORDER BY Total_Horses	List the total number of horses on farms in ascending order.	CREATE TABLE farm (Total_Horses VARCHAR)
SELECT Hosts FROM farm_competition WHERE Theme <> 'Aliens'	What are the hosts of competitions whose theme is not "Aliens"?	CREATE TABLE farm_competition (Hosts VARCHAR, Theme VARCHAR)

https://huggingface.co/datasets/b-mc2/sql-create-context/embed/viewer/default/train

DataFrame to Database




```python
# Database connection parameters for local PostgreSQL
db_name = ''            # Your database name
db_user = 'postgres'             # PostgreSQL default superuser
db_password = ''             # The password you set above
db_host = 'localhost'            # Localhost for local connection
db_port = '5432'                 # Default PostgreSQL port

# Create a connection string
connection_string = f'postgresql://{db_user}:{db_password}@{db_host}:{db_port}/{db_name}'

# Create a SQLAlchemy engine
engine = create_engine(connection_string)
```

```python
# Insert DataFrames into PostgreSQL tables
financial_statements_df.to_sql('financial_statements', engine, if_exists='replace', index=False)
online_shopping_df.to_sql('online_shopping', engine, if_exists='replace', index=False)
customer_support_tickets_df.to_sql('customer_support_tickets', engine, if_exists='replace', index=False)
spotify_data_df.to_sql('spotify_data', engine, if_exists='replace', index=False)
```

Download Database with SQLDatabase (langchain)

LangChain

```python
from langchain_community.utilities import SQLDatabase

db = SQLDatabase.from_uri(connection_string)
print(db.dialect)
print(db.get_usable_table_names())
```

```
postgresql
['customer_support_tickets', 'financial_statements', 'online_shopping', 'spotify_data']
```

```python
print(db.get_table_info())
```

```
CREATE TABLE customer_support_tickets (
        "Ticket_ID" BIGINT,
        "Customer_Name" TEXT,
        "Customer_Email" TEXT,
        "Customer_Age" BIGINT,
        "Customer_Gender" TEXT,
        "Product_Purchased" TEXT,
        "Date_of_Purchase" TEXT,
        "Ticket_Type" TEXT,
        "Ticket_Subject" TEXT,
        "Ticket_Description" TEXT,
        "Ticket_Status" TEXT,
        "Resolution" TEXT,
        "Ticket_Priority" TEXT,
        "Ticket_Channel" TEXT,
        "First_Response_Time" TEXT,
        "Time_to_Resolution" TEXT,
        "Customer_Satisfaction_Rating" DOUBLE PRECISION
)

/*
3 rows from customer_support_tickets table:
Ticket_ID       Customer_Name   Customer_Email  Customer_Age    Customer_Gender Product_Purchased       Date_of_Purchase
1       Marisa Obrien   carrollallison@example.com      32      Other   GoPro Hero      2021-03-22      Technical issue Product

Your billing zip code is: 71701.

        Pending Customer Response       None    Critical        Social media    2023-06-01 12:15:36     None    None
2       Jessica Rios    clarkeashley@example.com        42      Female  LG Smart TV     2021-05-22      Technical issue Periphe

If you need to change an existing       Pending Customer Response       None    Critical        Chat    2023-06-01 16:45:38
3       Christopher Robbins     gonzalestracy@example.com       48      Other   Dell XPS        2020-07-14      Technical issue
*/
```

Promp template for SQL Query

LangChain

```
from langchain_core.prompts import PromptTemplate

template = '''You are a PostgreSQL expert. Given an input question, first create a syntactically correct PostgreSQL query to run, then look at the results of the query and return the answer to the input question.
Unless the user specifies in the question a specific number of examples to obtain, query for at most 5 results using the LIMIT clause as per PostgreSQL. You can order the results to return the most informative data in the database.
Never query for all columns from a table. You must query only the columns that are needed to answer the question. Wrap each column name in double quotes (") to denote them as delimited identifiers.
Pay attention to use only the column names you can see in the tables below. Be careful to not query for columns that do not exist. Also, pay attention to which column is in which table.
Pay attention to use CURRENT_DATE function to get the current date, if the question involves "today".

Use the following format:

Question: Question here
SQLQuery: SQL Query to run

Answer only SQLQuery
Example :
    - SELECT COUNT(*) FROM head WHERE age > 56
    - SELECT MAX(budget_in_billions), MIN(budget_in_billions) FROM department
    - SELECT AVG(num_employees) FROM department WHERE ranking BETWEEN 10 AND 15

Only use the following tables:
{table_info}

Question: {input}'''
prompt = PromptTemplate.from_template(template, partial_variables={
    "table_info": db.get_table_info(),
    "top_k": 5,  # Default value for top_k
})
```

Promp template for SQL Query

LangChain

Step 1 SQLQuery from LLM

```
[ ] Quesion = "มี Ticket_ID จำนวนเท่าไหร่ ที่ Date_of_Purchase ตั้งแต่ 2020-02-01 ถึง 2020-02-28"
    response = chain.invoke({"question": Quesion})
    response
```

```
'Question: มี Ticket_ID จำนวนเท่าไหร่ ที่ Date_of_Purchase ตั้งแต่ 2020-02-01 ถึง 2020-02-28\nSQLQuery: SELECT COUNT("Ticket_ID") FROM customer_support_tickets WHERE "Date_of_Purchase" >= \'2020-02-01\' AND "Date_of_Purchase" <= \'2020-02-28\''
```

```
[ ] print(response.split('SQLQuery:')[-1])
```

```
 SELECT COUNT("Ticket_ID") FROM customer_support_tickets WHERE "Date_of_Purchase" >= '2020-02-01' AND "Date_of_Purchase" <= '2020-02-28'
```

Step 2 SQLResult from sqlquery

```
SQL_Result=db.run(response.split('SQLQuery:')[-1])
SQL_Result
```

```
'[(361,)]'
```

Step 3 get Answer from LLM

```
[ ] answer_prompt =f"""
    given the following user question, corresponding SQL query, and SQL result, answer the user question.

    Question: {Quesion}
    SQL Query: {response.split('SQLQuery:')[-1]}
    SQL Result: {SQL_Result}
    Answer:
    """
    messages = [
        ("system", answer_prompt),
    ]
    answer = llm.invoke(messages)
    answer
```

```
AIMessage(content='มี Ticket_ID จำนวน 361 รายการที่ Date_of_Purchase ตั้งแต่ 2020-02-01 ถึง 2020-02-28', response_metadata={'token_usage': {'completion_tokens': 35, 'prompt_tokens': 143, 'total_tokens': 178, 'completion_time': 0.14, 'prompt_time': 0.038098987, 'queue_time': 0.005538982999999997, 'total_time': 0.178098987}, 'model_name': 'llama-3.1-70b-versatile', 'system_fingerprint': 'fp_b6828be2c9', 'finish_reason': 'stop', 'logprobs': None}, id='run-0dc4505e-1766-41a2-8485-ba443a2d7081-0', usage_metadata={'input_tokens': 143, 'output_tokens': 35, 'total_tokens': 178})
```

```
[ ] answer.content
```

```
'มี Ticket_ID จำนวน 361 รายการที่ Date_of_Purchase ตั้งแต่ 2020-02-01 ถึง 2020-02-28'
```

Example Context

langchain_community.utilities import SQLDatabase

CONTEXT

```
CREATE TABLE financial_statements (
        "Year" BIGINT,
        "Company" TEXT,
        "Category" TEXT,
        "Market Cap(in B USD)" DOUBLE PRECISION,
        "Revenue" DOUBLE PRECISION,
        "Gross Profit" DOUBLE PRECISION,
        "Net Income" DOUBLE PRECISION,
        "Earning Per Share" DOUBLE PRECISION,
        "EBITDA" DOUBLE PRECISION,
        "Share Holder Equity" DOUBLE PRECISION,
        "Cash Flow from Operating" DOUBLE PRECISION,
        "Cash Flow from Investing" DOUBLE PRECISION,
        "Cash Flow from Financial Activities" DOUBLE PRECISION,
        "Current Ratio" DOUBLE PRECISION,
        "Debt/Equity Ratio" DOUBLE PRECISION,
        "ROE" DOUBLE PRECISION,
        "ROA" DOUBLE PRECISION,
        "ROI" DOUBLE PRECISION,
        "Net Profit Margin" DOUBLE PRECISION,
        "Free Cash Flow per Share" DOUBLE PRECISION,
        "Return on Tangible Equity" DOUBLE PRECISION,
        "Number of Employees" BIGINT,
        "Inflation Rate(in US)" DOUBLE PRECISION
)

/*
3 rows from financial_statements table:
Year    Company Category        Market Cap(in B USD)    Revenue Gross Profit    Net Income      Earning Per Share       EBITDA  Share Holder Equity     Cash Flow f
2022    AAPL    IT      2066.94 394328.0        170782.0        99803.0 6.11    130541.0        50672.0 122151.0        -22354.0        -110749.0       0.8794  2.3
2021    AAPL    IT      2913.28 365817.0        152836.0        94680.0 5.61    120233.0        63090.0 104038.0        -14545.0        -93353.0        1.0746  1.9
2020    AAPL    IT      2255.97 274515.0        104956.0        57411.0 3.28    77344.0 65339.0 80674.0 -4289.0 -86820.0        1.3636  1.7208  87.8664 17.7256 35.
*/
```

QESTION

Question: บริษัท AAPL จัดอยู่ใน Category อะไร ในปี 2022

SQL CODE

SQL Query: SELECT "Category" FROM financial_statements WHERE "Company" = 'AAPL' AND "Year" = 2022

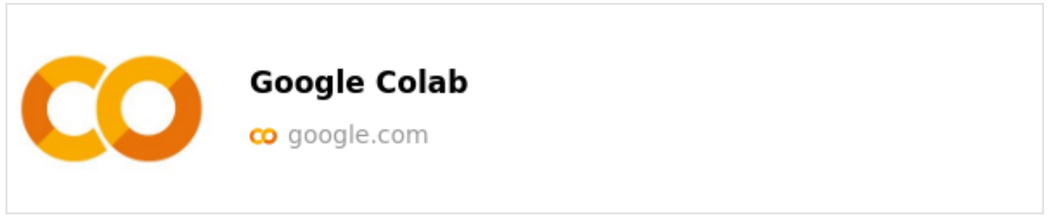
SQL RESULT

SQL Result: [('IT',)]

LLM ANSWER

Answer:
บริษัท AAPL จัดอยู่ใน Category 'IT' ในปี 2022

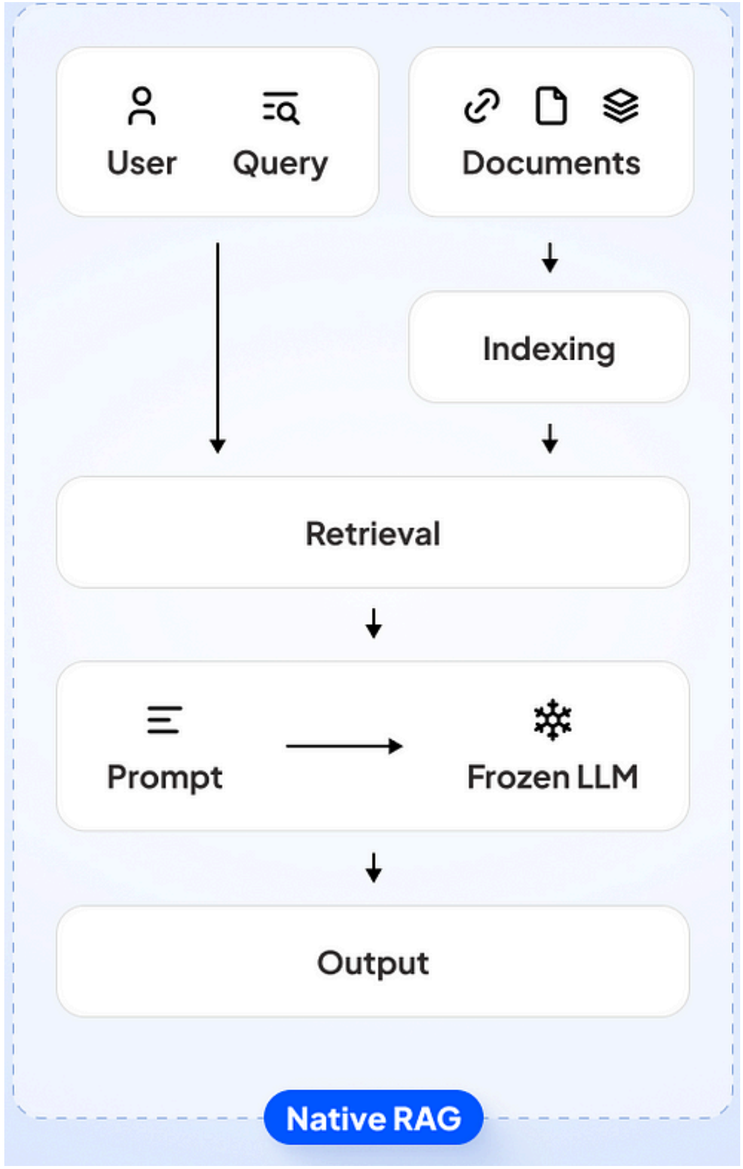
Workshop



https://colab.research.google.com/drive/1BEST7HrkA5MDJjknhCZ0Hv0IMZZ8B1Su?usp=sharing

UNSTRUCTURED RAG

Native RAG


User
Query
Documents
Indexing
Retrieval
Prompt
Frozen LLM
Output
Native RAG

Similarity searches

- **Dense Vector Search (embedding)**
- **Sparse Vector Search (TF/IDF)**

Feature	Sparse Vectors	Dense Vectors
Data Representation	Majority of elements are zero	All elements are non-zero
Computational Efficiency	Generally higher, especially in operations involving zero elements	Lower, as operations are performed on all elements
Information Density	Less dense, focuses on key features	Highly dense, capturing nuanced relationships
Example Applications	Text search, Hybrid search	RAG, many general machine learning tasks

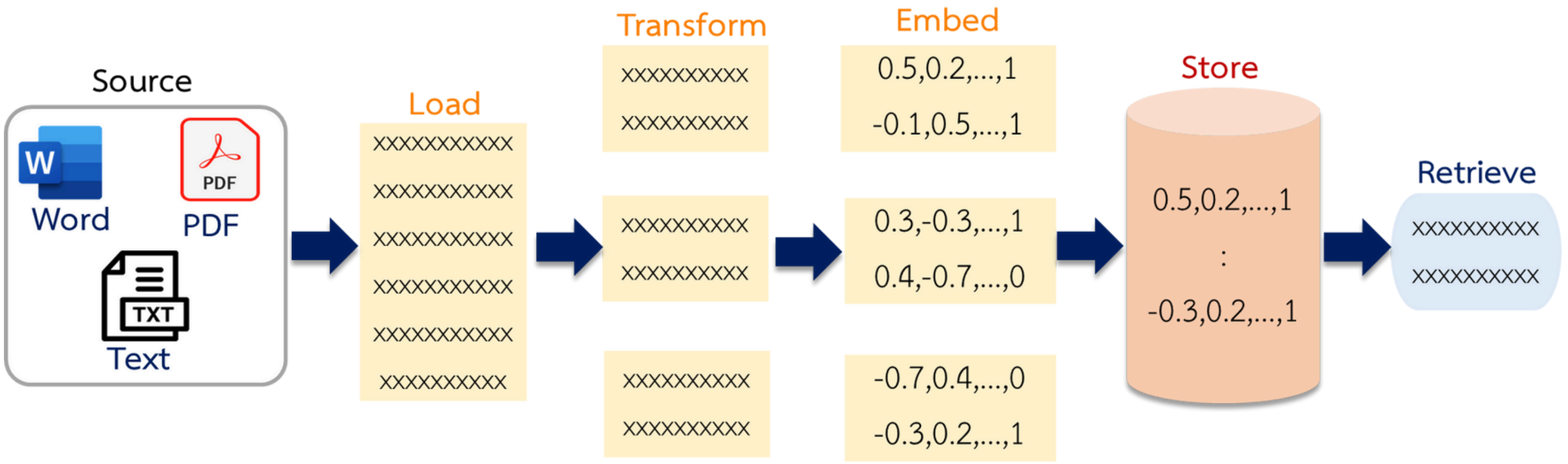
dense

```
[0.14, 1.72, ..., -0.94]
```

sparse

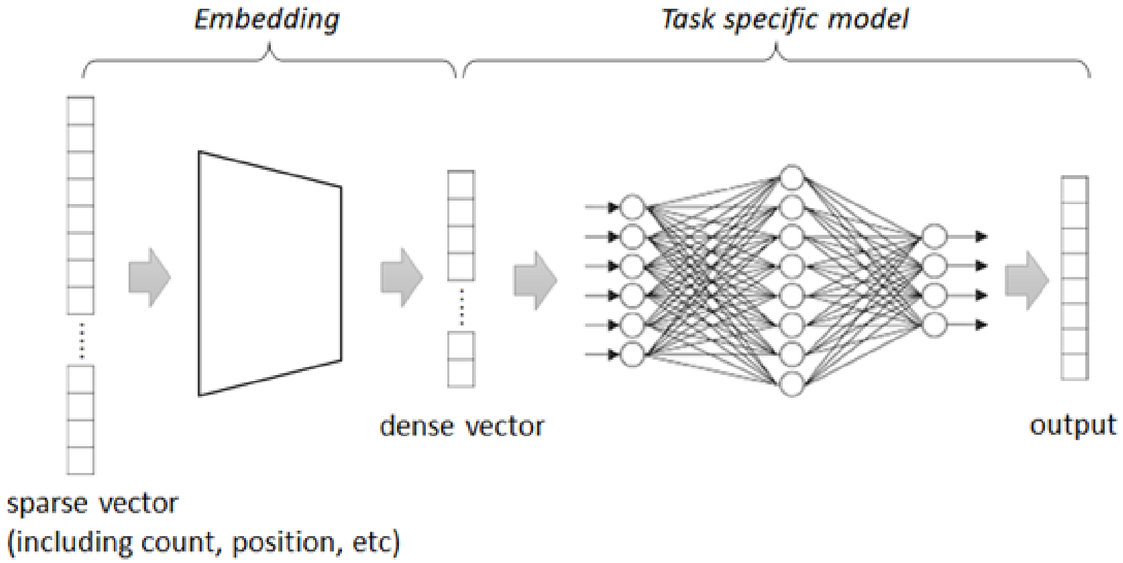
```
{
  "The": 0.124,
  "quick": 9.131,
  "brown": 7.138,
  "fox": 4.972,
  "jumps": 7.421,
  "over": 1.014,
  "the": 0.124,
  "lazy": 3.916,
  "dog": 1.532
}
```

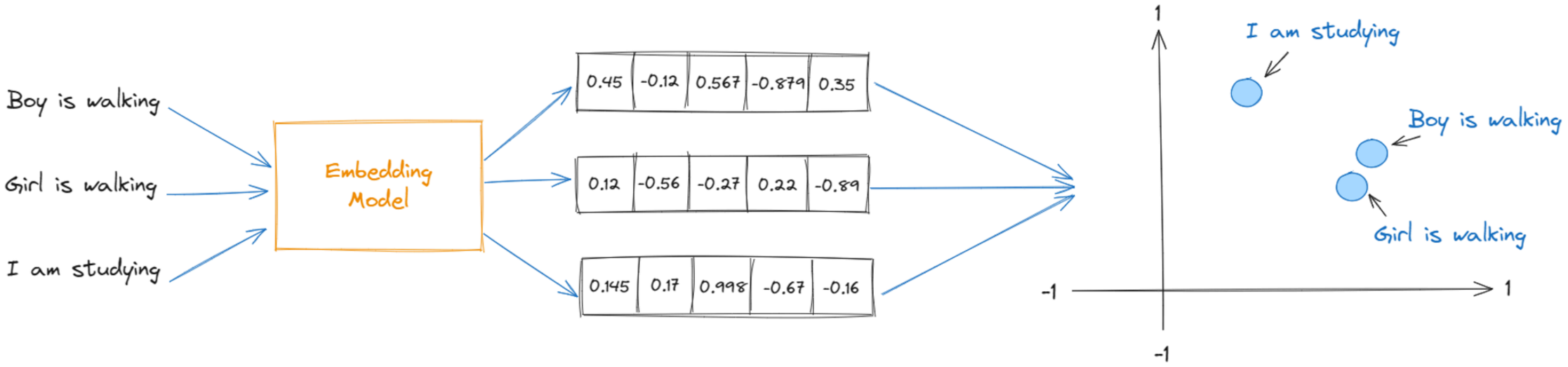
Dense Vector Search RAG



1. **loader:** loading of documents from various sources
2. **Transformers:** Split the text up into small, semantically meaningful chunks
3. **Embedding models**: convert text into a list of floating point numbers
4. **Vector stores**: Store and search over embedded data
5. **Retrievers**: querying the stored data

Dense Vector Search





Thai Sentence Vector Benchmark

XQuAD

Base Model	R@1	MRR@10	Supervised?	Latency(second)
simcse-model-distil-m-bert	18.24	27.19		0.61
simcse-model-m-bert-thai-cased	22.94	30.29		1.02
simcse-model-XLMR	52.02	62.94		0.85
simcse-model-wangchanberta	53.87	65.51		0.81
simcse-model-phayathaibert	73.95	81.67		0.79
SCT-model-XLMR	55.29	65.23		1.24
SCT-model-wangchanberta	66.30	76.14		1.23
SCT-model-phayathaibert	67.56	76.14		1.19
SCT-Distil-model-XLMR	68.91	78.19		1.24
SCT-Distil-model-wangchanberta	62.27	72.53		1.35
SCT-Distil-model-phayathaibert	71.43	80.18		1.21
SCT-Distil-model-phayathaibert-bge-m3	80.50	86.75		
ConGen-model-XLMR	71.76	80.01		1.24
ConGen-model-wangchanberta	70.92	79.59		1.21
ConGen-model-phayathaibert	71.85	80.33		1.19
ConGen-BGE_M3-model-phayathaibert	85.80	90.48		1.3
distiluse-base-multilingual-cased-v2	49.16	58.19	✓	1.05
paraphrase-multilingual-mpnet-base-v2	71.26	79.63	✓	1.24
BGE M-3	90.50	94.33	✓	7.22
Cohere-embed-multilingual-v2.0	82.52	87.78	✓	XXX

TyDiQA

Base Model	R@1	MRR@10	Supervised?	Latency(second)
simcse-model-distil-m-bert	44.69	51.39		1.6
simcse-model-m-bert-thai-cased	45.09	52.37		2.46
simcse-model-XLMR	58.06	64.72		2.35
simcse-model-wangchanberta	62.65	70.02		2.32
simcse-model-phayathaibert	71.43	78.16		2.28
SCT-model-XLMR	49.28	58.62		3.15
SCT-model-wangchanberta	58.19	68.05		3.21
SCT-model-phayathaibert	63.43	71.73		3.21
SCT-Distil-model-XLMR	56.36	65.18		3.3
SCT-Distil-model-wangchanberta	56.23	65.18		3.18
SCT-Distil-model-phayathaibert	58.32	67.42		3.21
SCT-Distil-model-phayathaibert-bge-m3	78.37	84.01		
ConGen-model-XLMR	60.29	68.56		3.28
ConGen-model-wangchanberta	59.11	67.42		3.19
ConGen-model-phayathaibert	59.24	67.69		3.15
ConGen-BGE_M3-model-phayathaibert	83.36	88.29		3.14
distiluse-base-multilingual-cased-v2	32.50	42.20	✓	2.05
paraphrase-multilingual-mpnet-base-v2	54.39	63.12	✓	3.16
BGE M-3	89.12	93.43	✓	20.87
Cohere-embed-multilingual-v2.0	85.45	90.33	✓	XXX

https://github.com/mrpeerat/Thai-Sentence-Vector-Benchmark

Suggestion embedding model

BAAI/bge-m3 · Hugging Face

We’re on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface

jinaai/jina-embeddings-v3 · Hugging Face

We’re on a journey to advance and democratize artificial intelligence through open source and open science.

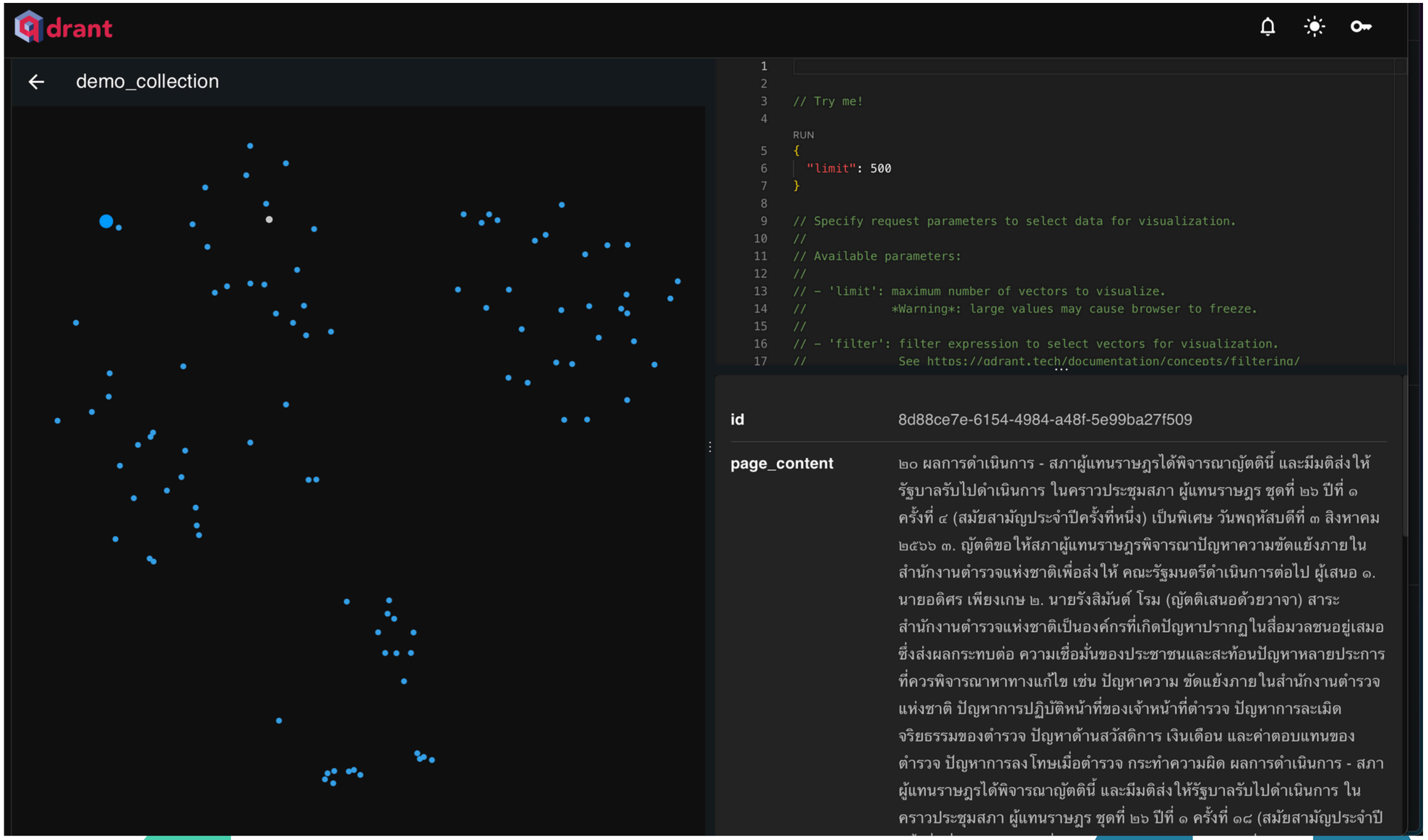
huggingface

Vectorstore

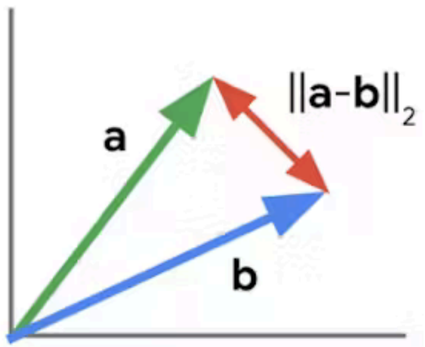
Vectorstore	Delete by ID	Filtering	Search by Vector	Search with score	Async	Passes Standard Tests	Multi Tenancy	IDs in a Docume
AstraDBVectorStore	✅	✅	✅	✅	✅	❌	❌	❌
Chroma	✅	✅	✅	✅	✅	❌	❌	❌
Clickhouse	✅	✅	❌	✅	❌	❌	❌	❌
CouchbaseVectorStore	✅	✅	❌	✅	✅	❌	❌	❌
DatabricksVectorSearch	✅	✅	✅	✅	✅	❌	❌	❌
ElasticsearchStore	✅	✅	✅	✅	✅	❌	❌	❌
FAISS	✅	✅	✅	✅	✅	❌	❌	❌
InMemoryVectorStore	✅	✅	❌	✅	✅	❌	❌	❌
Milvus	✅	✅	❌	✅	✅	❌	❌	❌
MongoDBAtlasVectorSearch	✅	✅	✅	✅	✅	❌	❌	❌
PGVector	✅	✅	✅	✅	✅	❌	❌	❌
PineconeVectorStore	✅	✅	✅	❌	✅	❌	❌	❌
QdrantVectorStore	✅	✅	✅	✅	✅	❌	❌	❌
Redis	✅	✅	✅	✅	✅	❌	❌	❌
Weaviate	✅	✅	✅	✅	✅	❌	✅	❌

https://python.langchain.com/docs/integrations/vectorstores/

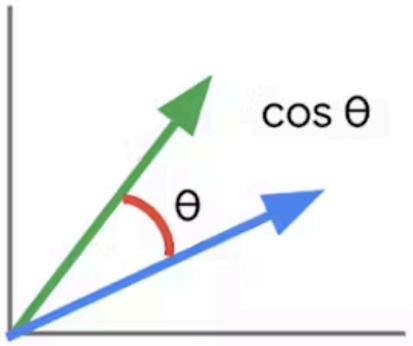
Vector Store database



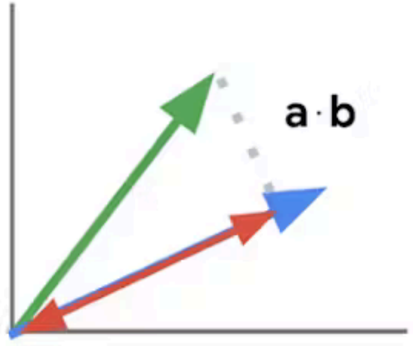
Vector Similarity



L2 distance

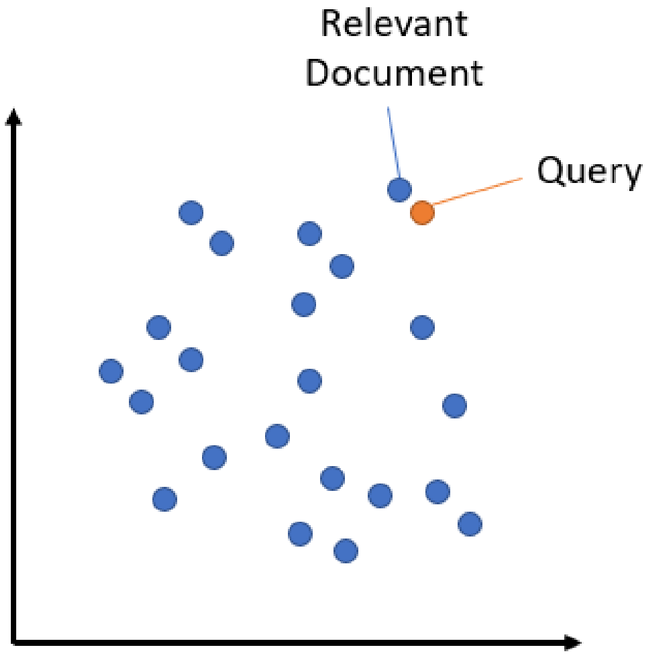


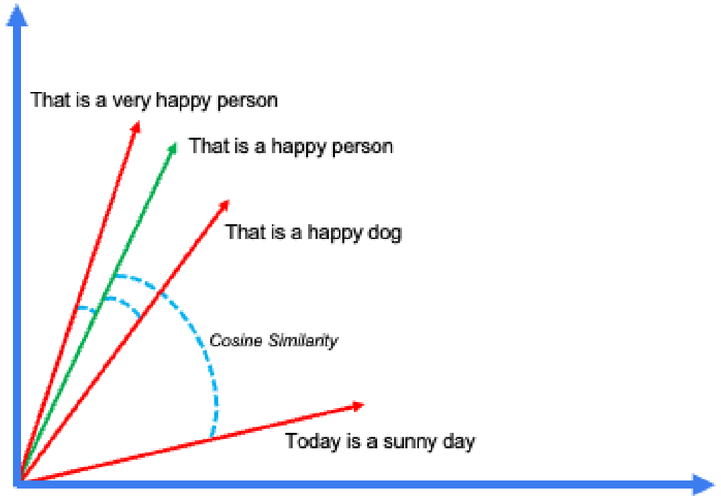
cosine similarity



inner product

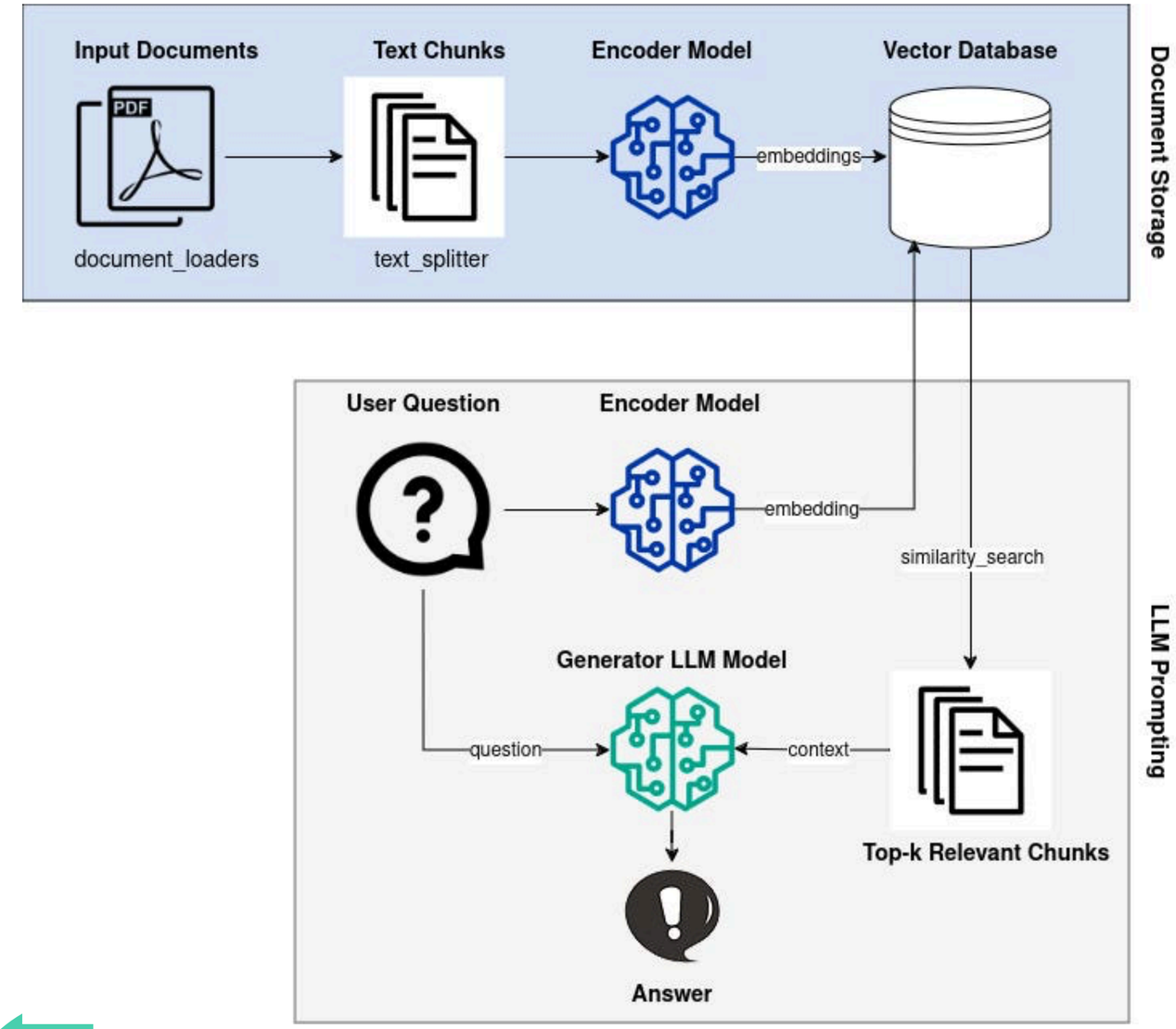
Calculating vector similarity



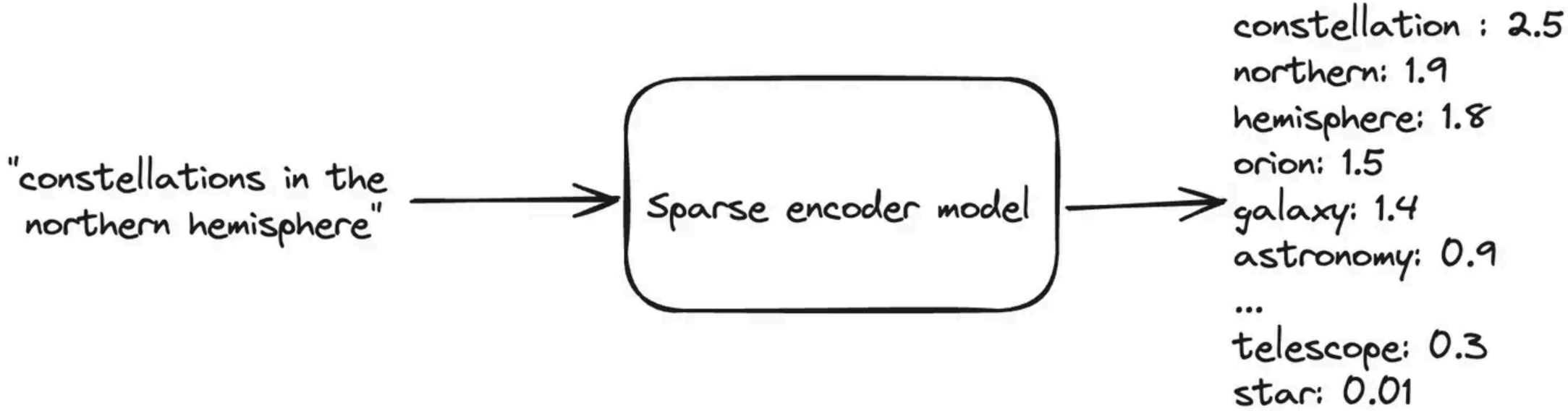


RAG

Retrieval-Augmented Generation



Sparse Vector Search



BM25 (Best Match 25)

What is BM25

BM25, or Best Match 25, also known as Okapi BM25, is a ranking algorithm for information retrieval and search engines that determines a document's relevance to a given query and ranks documents based on their relevance scores.

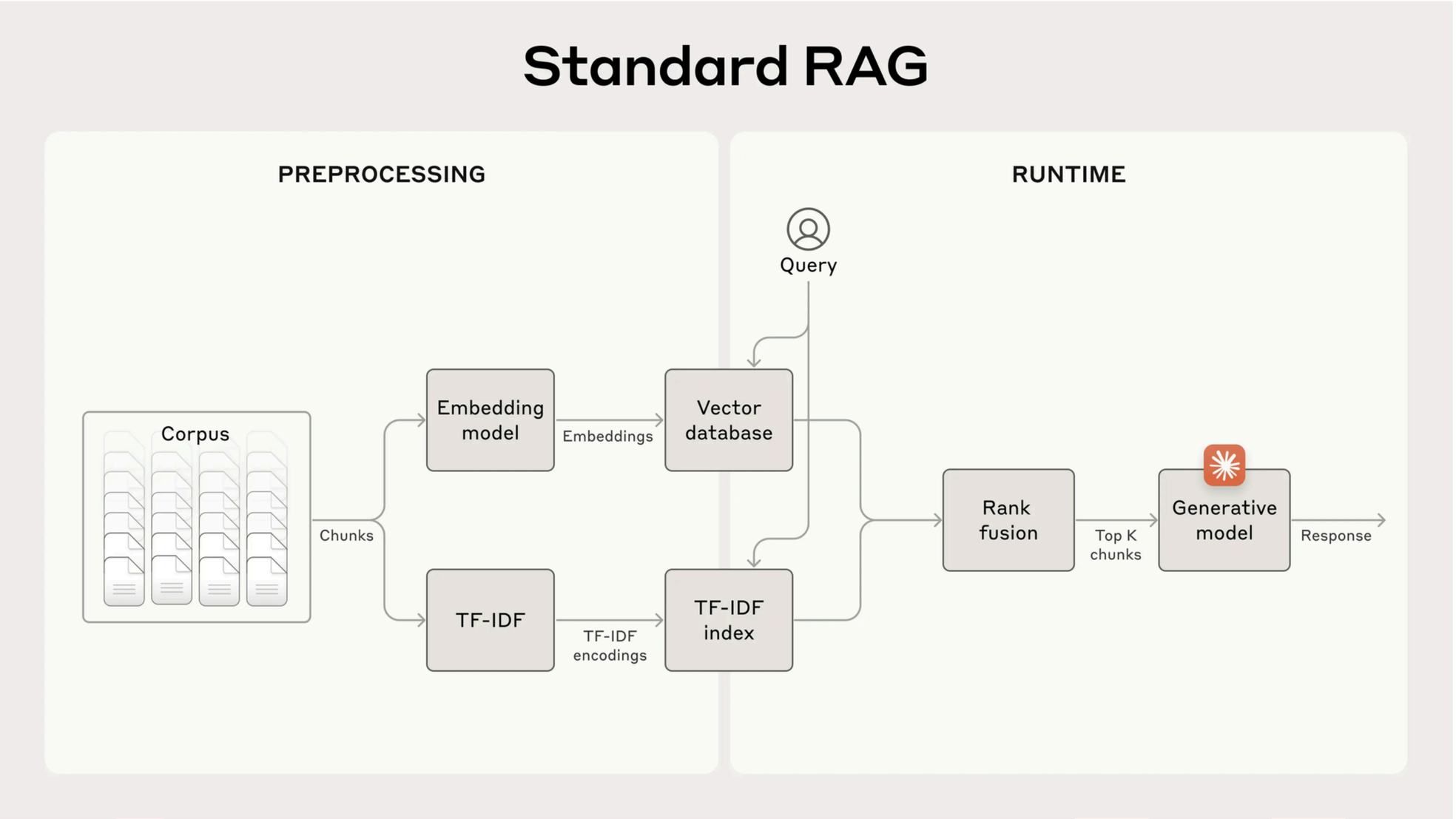
How does BM25 work?

The BM25 retrieval function calculates a relevance score for each document based on a specific search query.

The algorithm looks at three things:

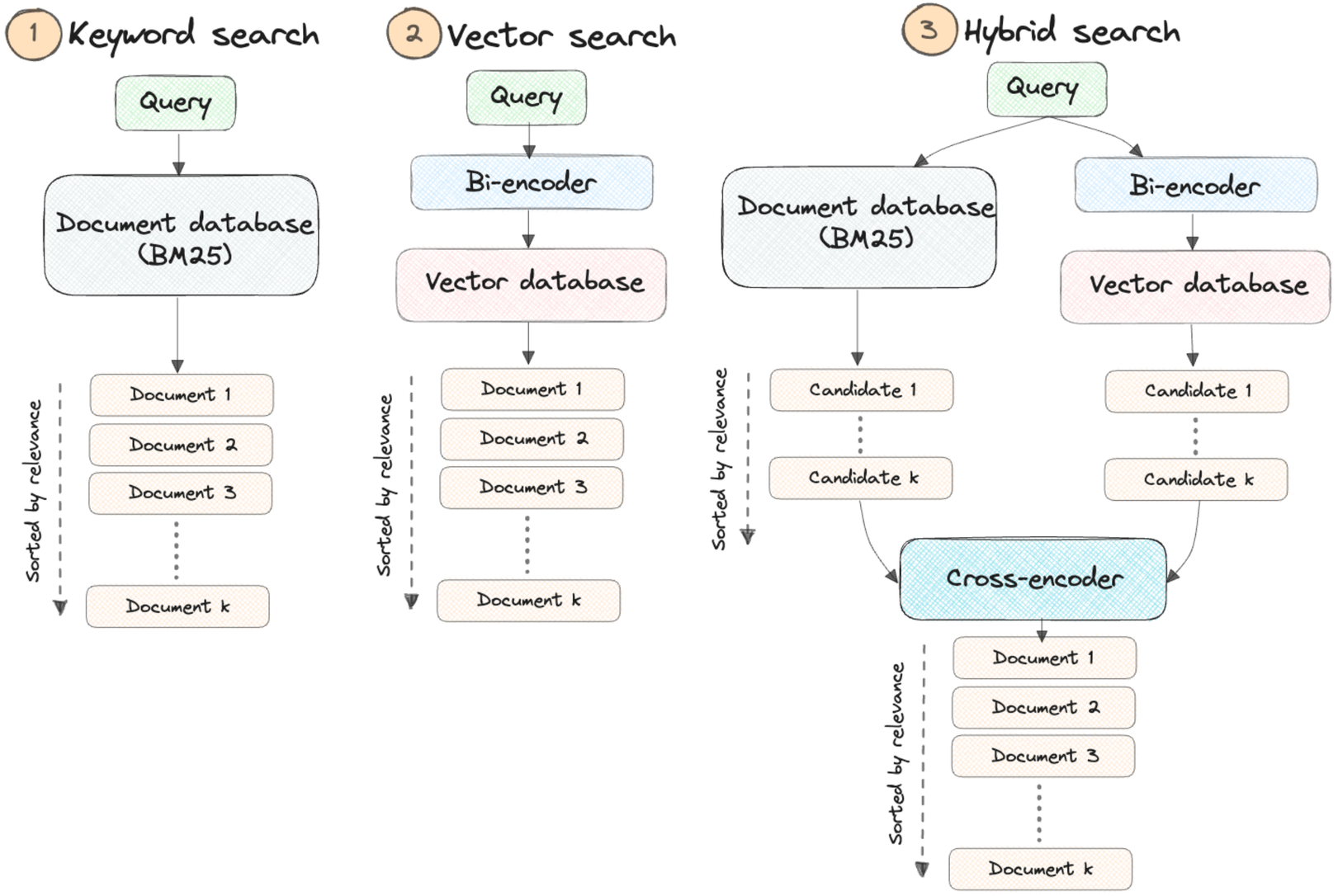
1. How often do the query terms appear in the document.
2. The length of the document.
3. The average length of all documents in the collection.

Standard RAG

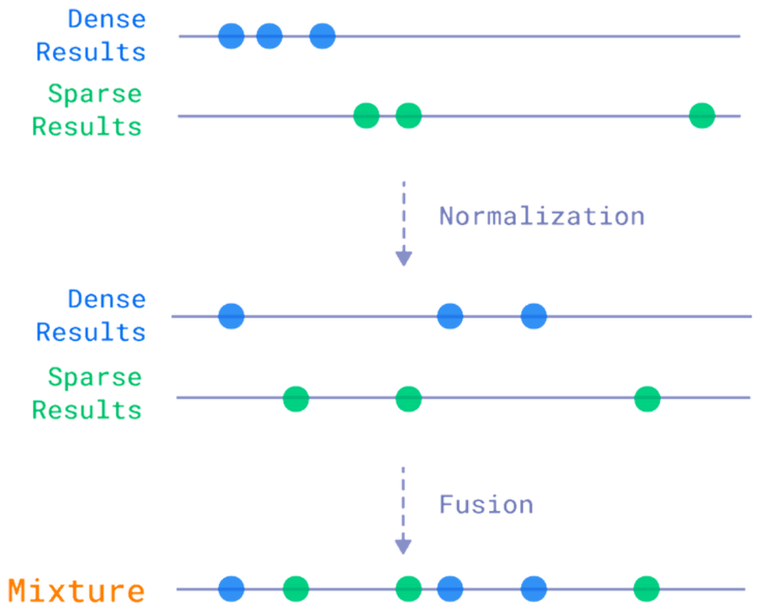


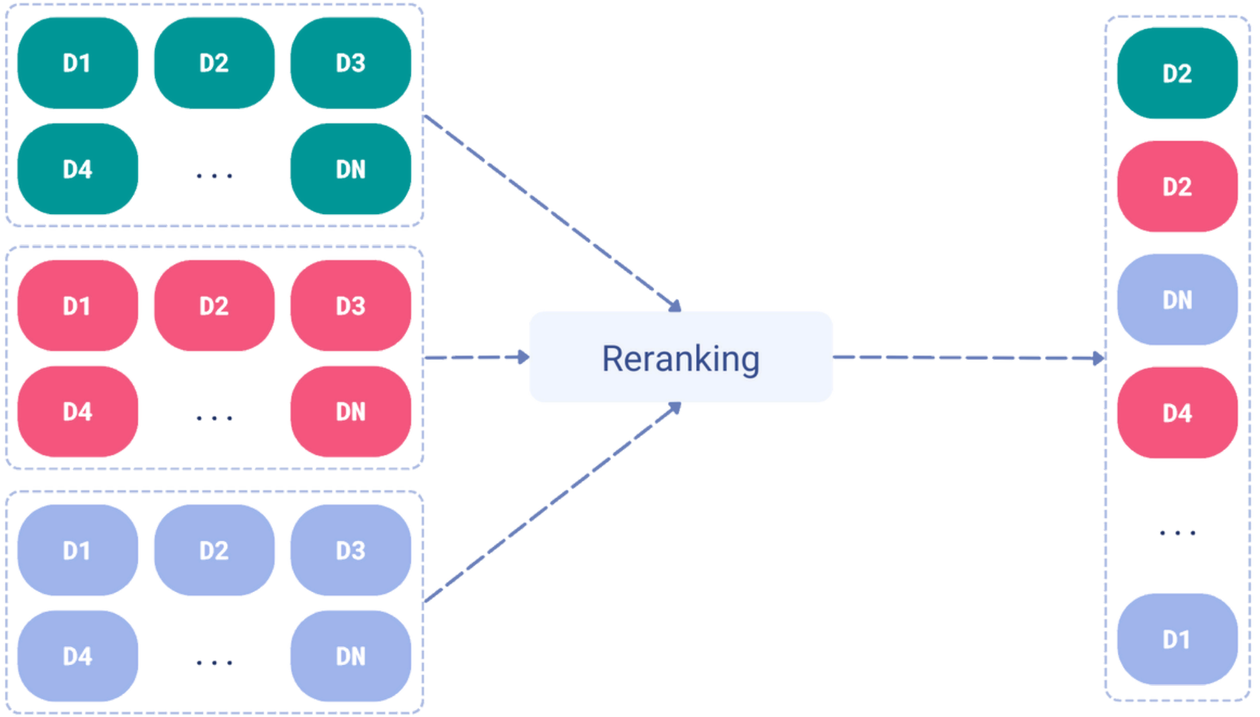
https://www.anthropic.com/news/contextual-retrieval

Hybrid Search

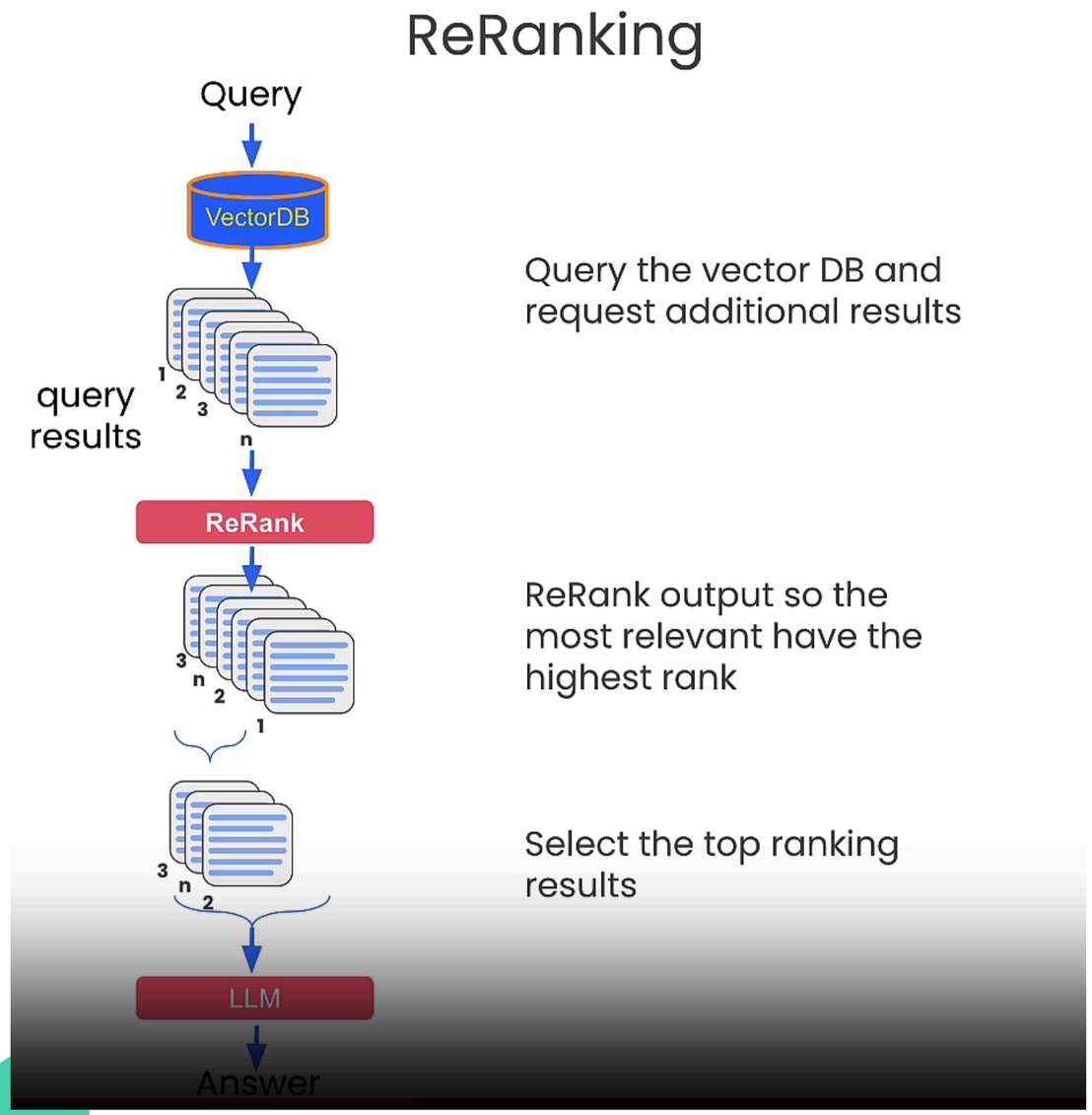


Rerank


Dense Results
Sparse Results
Normalization
Dense Results
Sparse Results
Fusion
Mixture


D1
D2
D3
D4
...
DN
D1
D2
D3
D4
...
DN
D1
D2
D3
D4
...
DN
Reranking
D2
D2
DN
D4
...
D1

Rerank

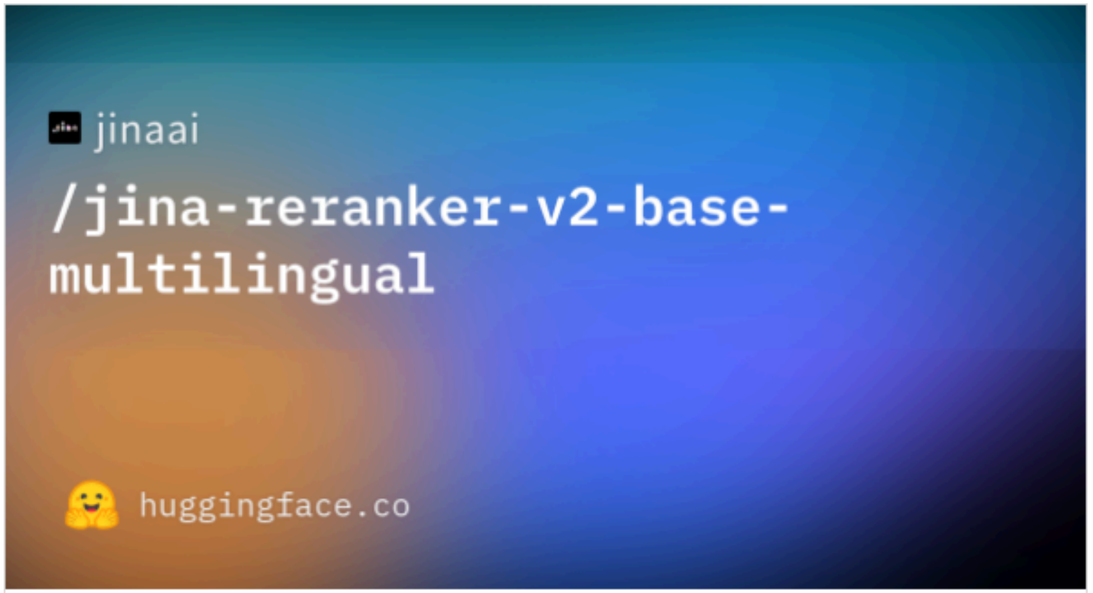

ReRanking
Query
VectorDB
query results
1
2
3
n
ReRank
3
n
2
1
3
n
2
LLM
Answer
Query the vector DB and request additional results
ReRank output so the most relevant have the highest rank
Select the top ranking results

Rerank

BAAI/bge-reranker-v2-m3 · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface

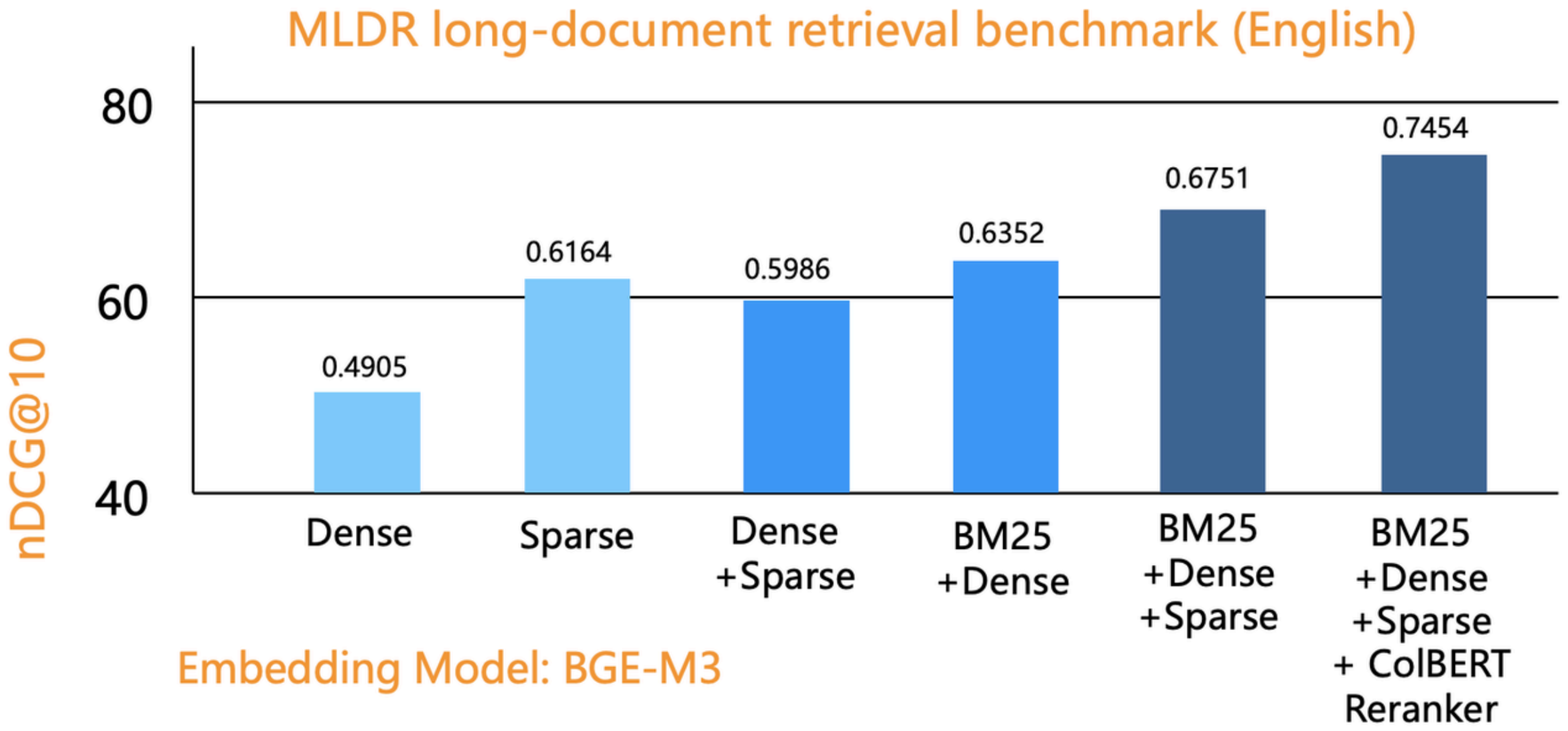


jinaai/jina-reranker-v2-base-multilingual · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

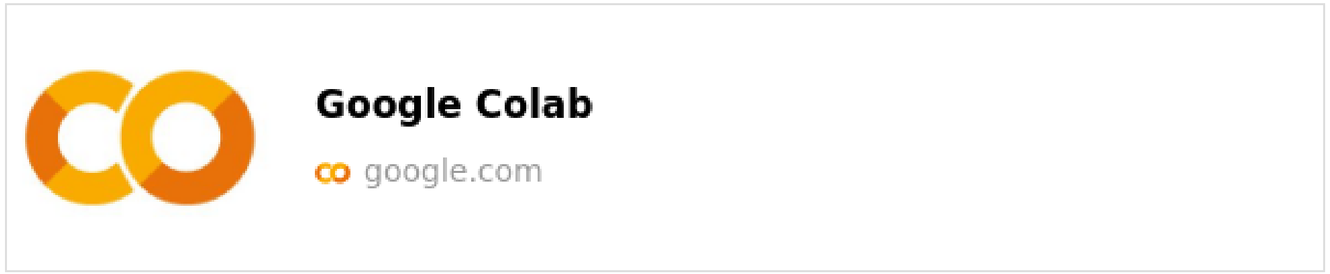
huggingface

Hybrid Search



https://infiniflow.org/blog/best-hybrid-search-solution

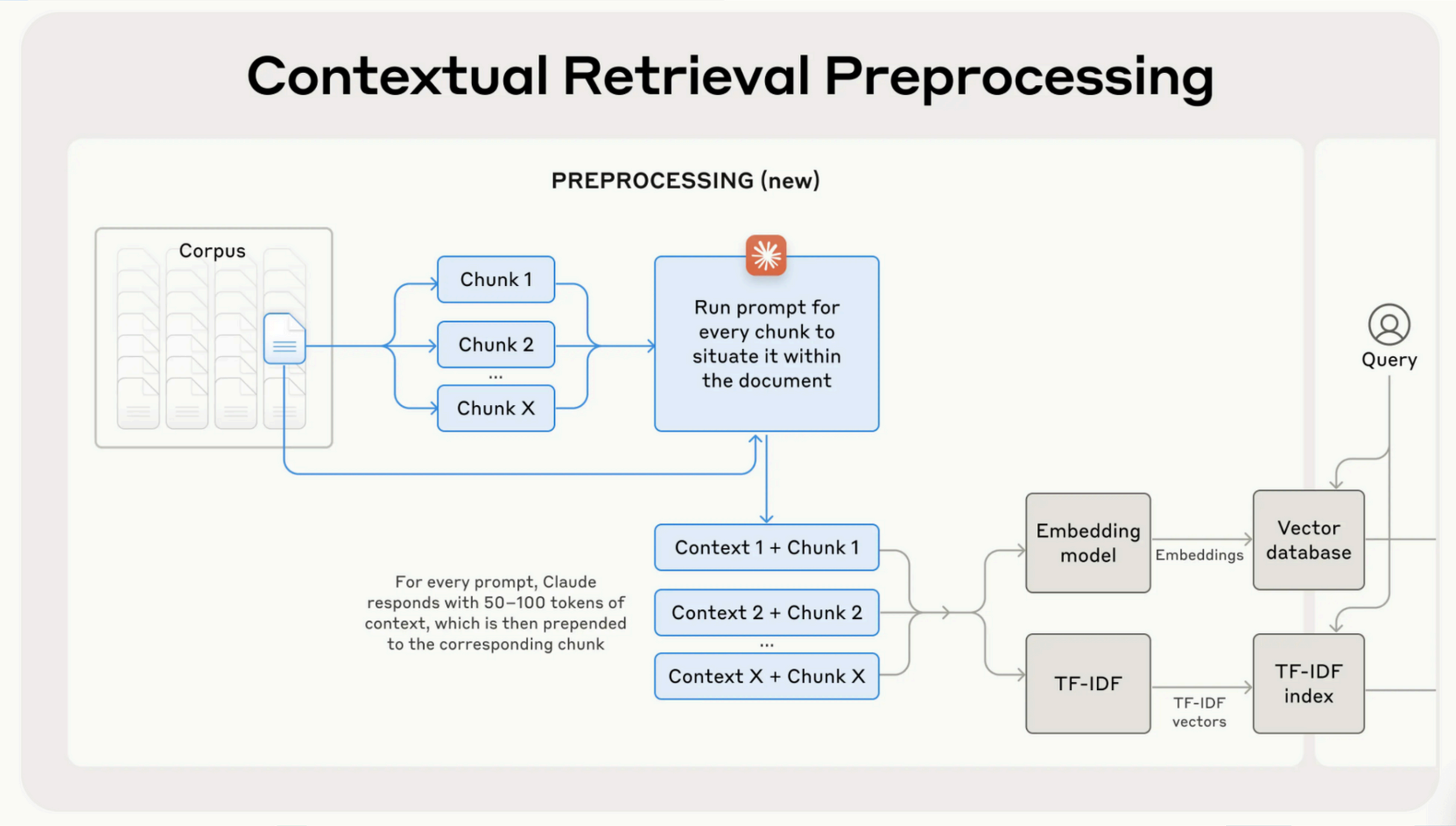
Workshop



https://colab.research.google.com/drive/1-zDPepPcHntU4riY8FBYr6ot8VKtn0M5?usp=sharing

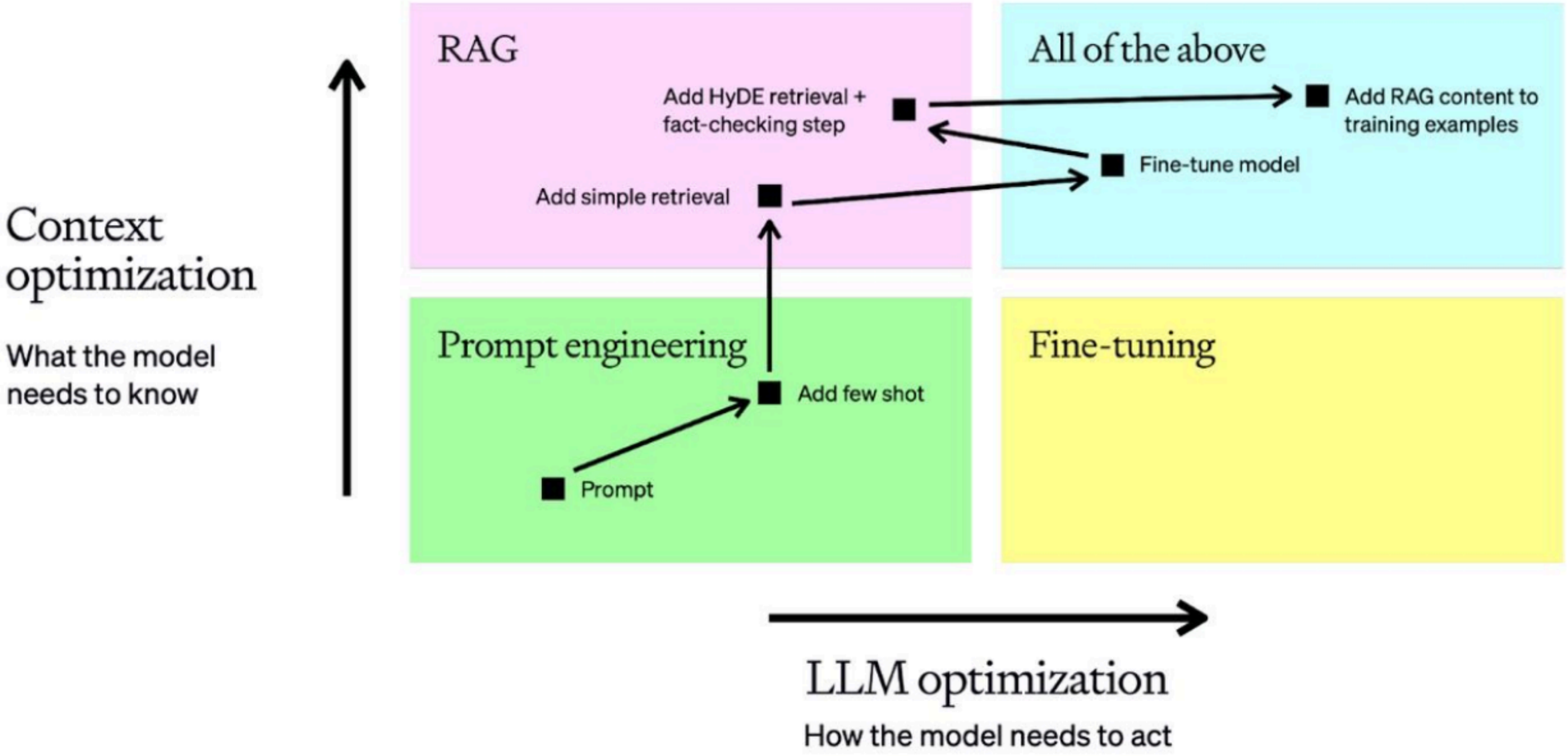
ADDITION RAG TECHNIQUE

Contextual Retrieval



https://www.anthropic.com/news/contextual-retrieval

LLM optimization context



HyDE

	DL19			DL20		
	map	ndcg@10	recall@1k	map	ndcg@10	recall@1k
w/o relevance judgement						
BM25	30.1	50.6	75.0	28.6	48.0	78.6
Contriever	24.0	44.5	74.6	24.0	42.1	75.4
HyDE	**41.8**	**61.3**	**88.0**	**38.2**	**57.9**	**84.4**
w/ relevance judgement						
DPR	36.5	62.2	76.9	41.8	**65.3**	81.4
ANCE	37.1	**64.5**	75.5	40.8	64.6	77.6
Contriever^{FT}	41.7	62.1	83.6	**43.6**	63.2	**85.8**

Table 1: Results for web search on DL19/20. Best performing w/o relevance and overall system(s) are marked **bold**. DPR, ANCE and Contriever^{FT} are in-domain *supervised* models that are finetuned on MS MARCO training data.

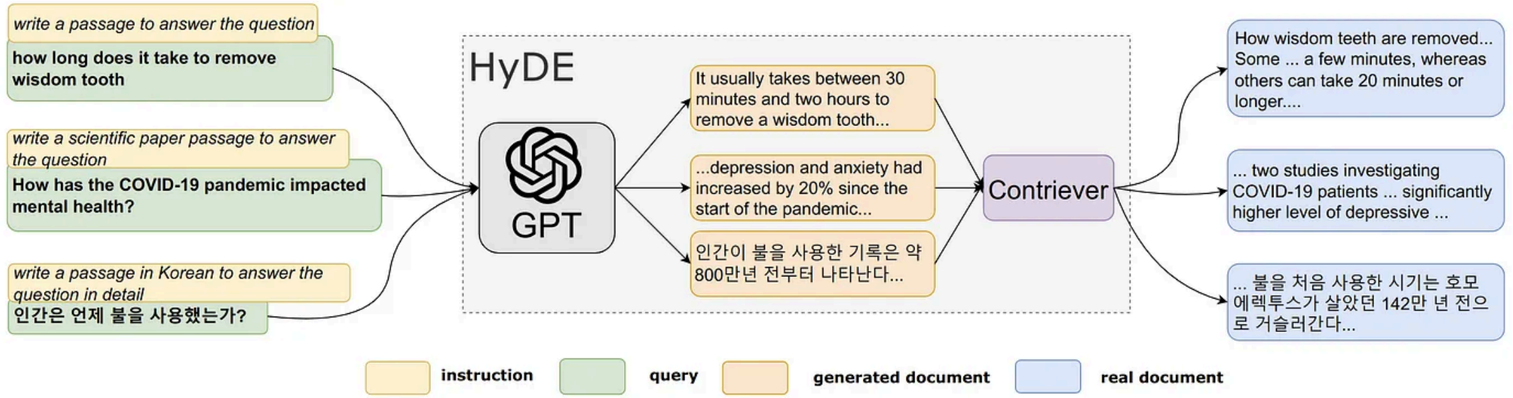
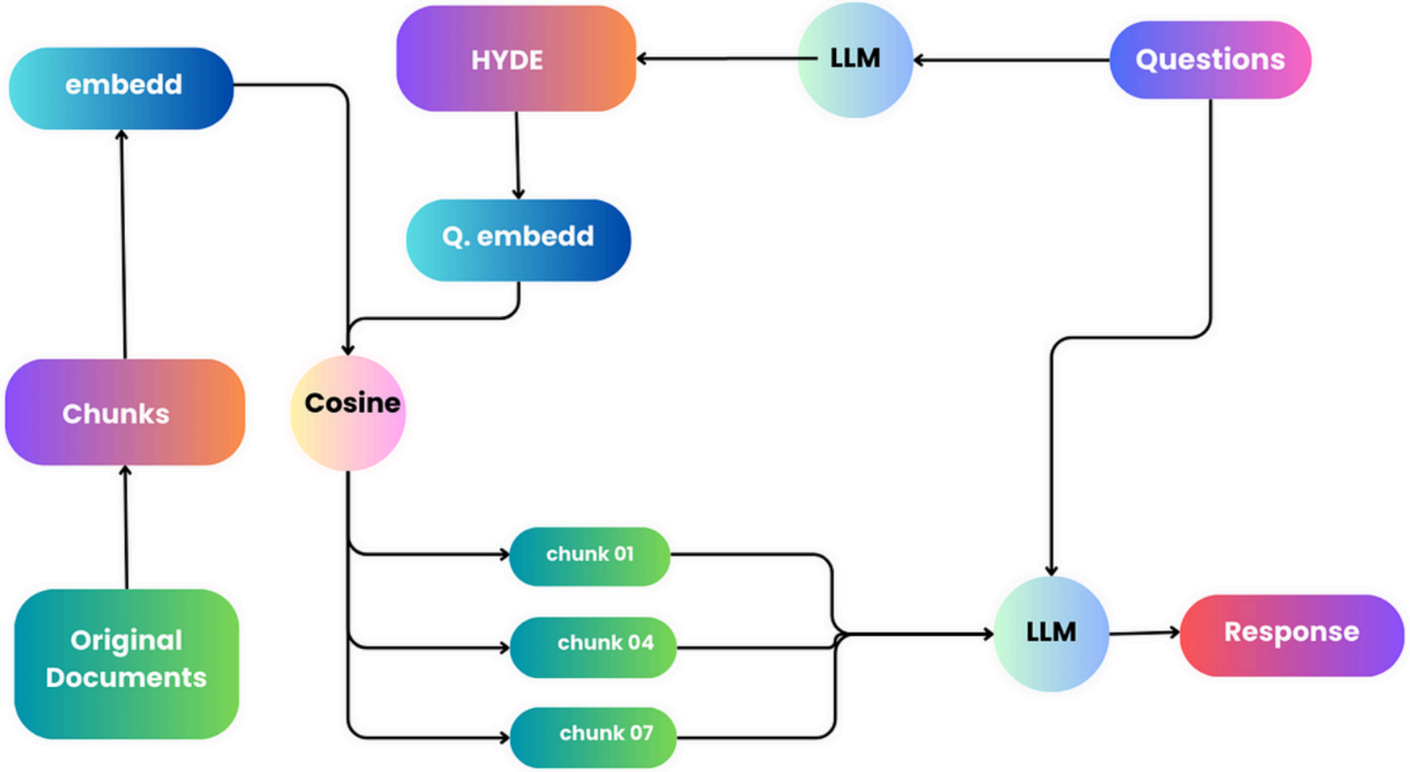


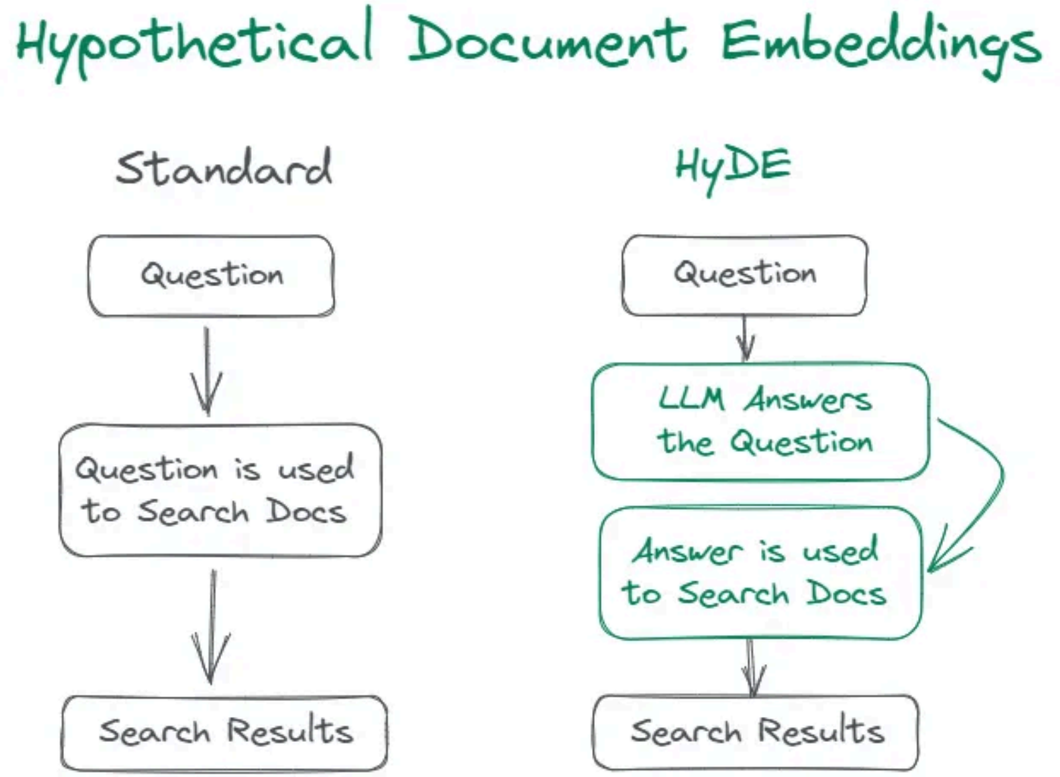
Figure 1: An illustration of the HyDE model. Documents snippets are shown. HyDE serves all types of queries without changing the underlying GPT-3 and Contriever/mContriever models.

https://arxiv.org/pdf/2212.10496

https://medium.aiplanet.com/advanced-rag-improving-retrieval-using-hypothetical-document-embeddings-hyde-1421a8ec075a

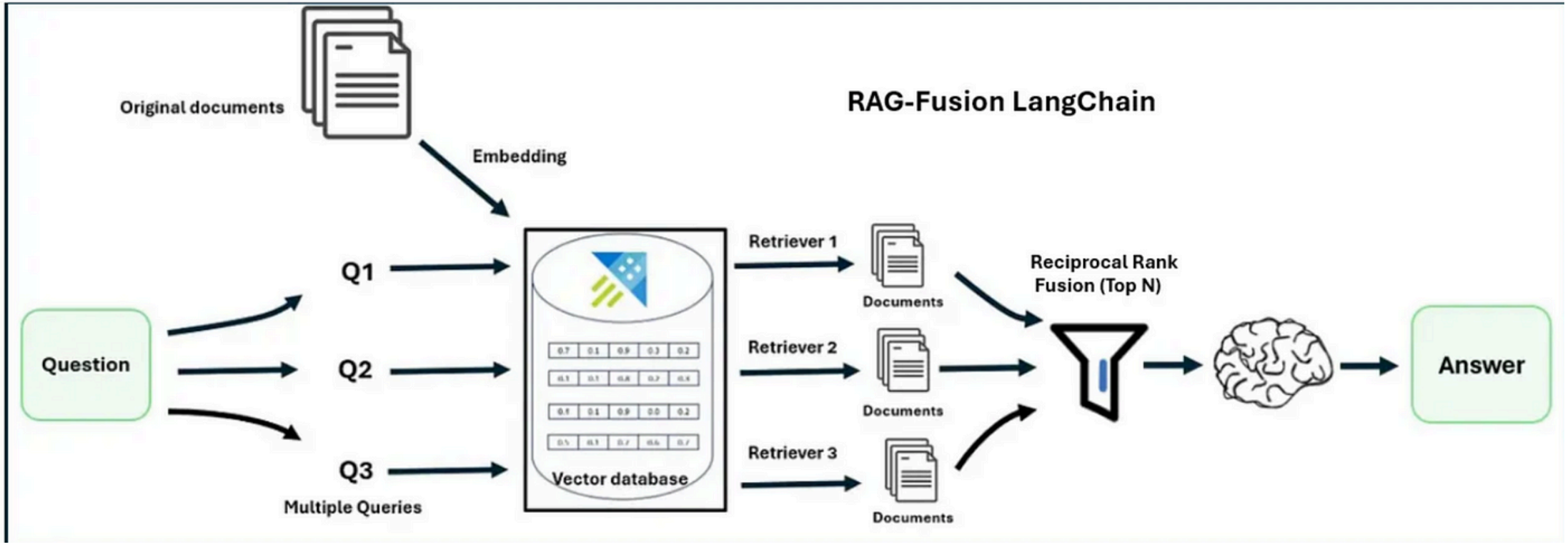
HyDE





https://medium.aiplanet.com/advanced-rag-improving-retrieval-using-hypothetical-document-embeddings-hyde-1421a8ec075a

Fusion RAG



https://medium.com/@nageshmashette32/langchain-rag-fusion-advance-rag-32eefc63da99

THANK YOU!

Get in Touch With Us