



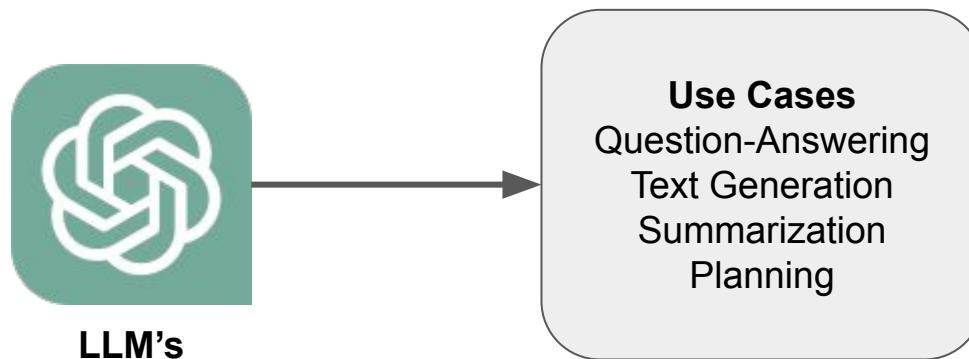
RAG with LlamaIndex

Ravi Theja
Developer Advocate, LlamaIndex

https://github.com/jerryjliu/llama_index

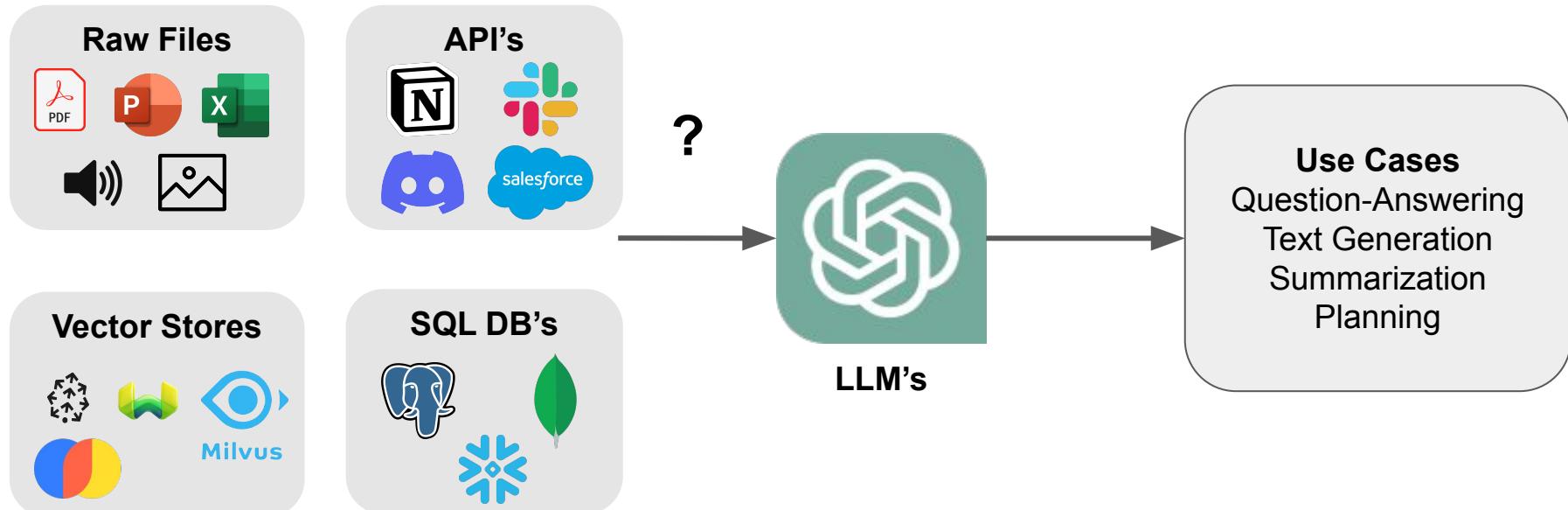
Context

- LLMs are a phenomenal piece of technology for knowledge generation and reasoning. They are pre-trained on large amounts of **publicly available data**.



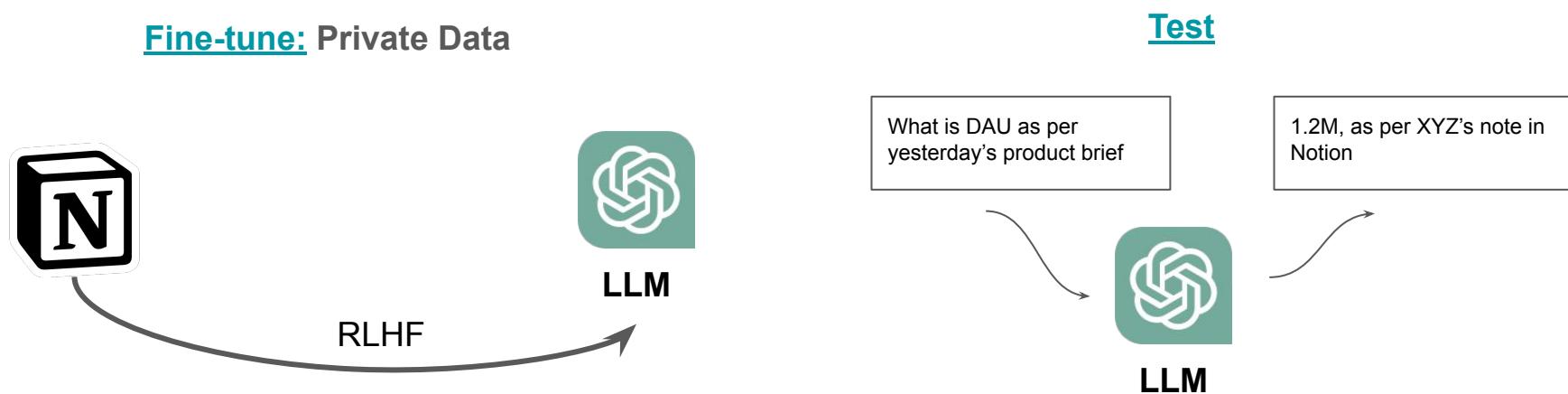
Context

- How do we best augment LLMs with our own **private data**?



Paradigms for inserting knowledge

Fine-tuning - baking knowledge into the weights of the network



Paradigms for inserting knowledge

Fine-tuning - baking knowledge into the weights of the network

Downsides:

- Data preparation effort
- Doesn't work well
- High upfront cost
- Online learning/ Updating new data - hard.

Paradigms for inserting knowledge

In-context learning - Fix the model, put context into the prompt

During yesterday's product brief meeting, the team discussed several key metrics, with a focus on Daily Active Users (DAU). Karthik provided an update on the DAU, highlighting that it reached 1.2M, marking a 3% growth compared to last week. The team attributed this growth to the recent updates in the onboarding process and expressed optimism for continued upward trends.....

Test

What is DAU as per yesterday's product brief, given following docs

1.2M, as per XYZ's note in Notion



LLM

Prod Brief
Meeting Notes

Paradigms for inserting knowledge

In-context learning - Fix the model, put context into the prompt

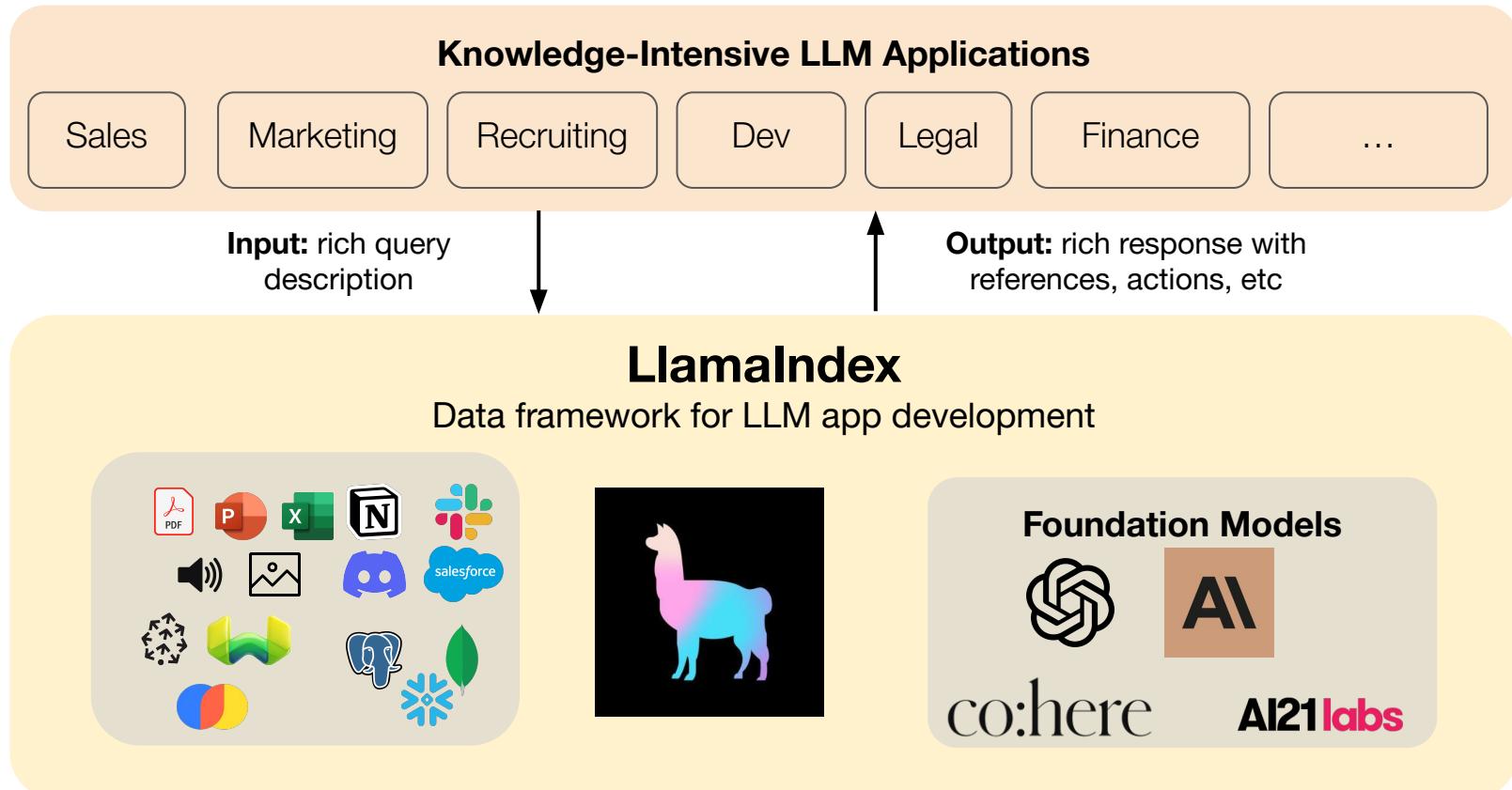
Downsides:

- Model token limit on context length.
- Right context might

Key challenges of in-context learning

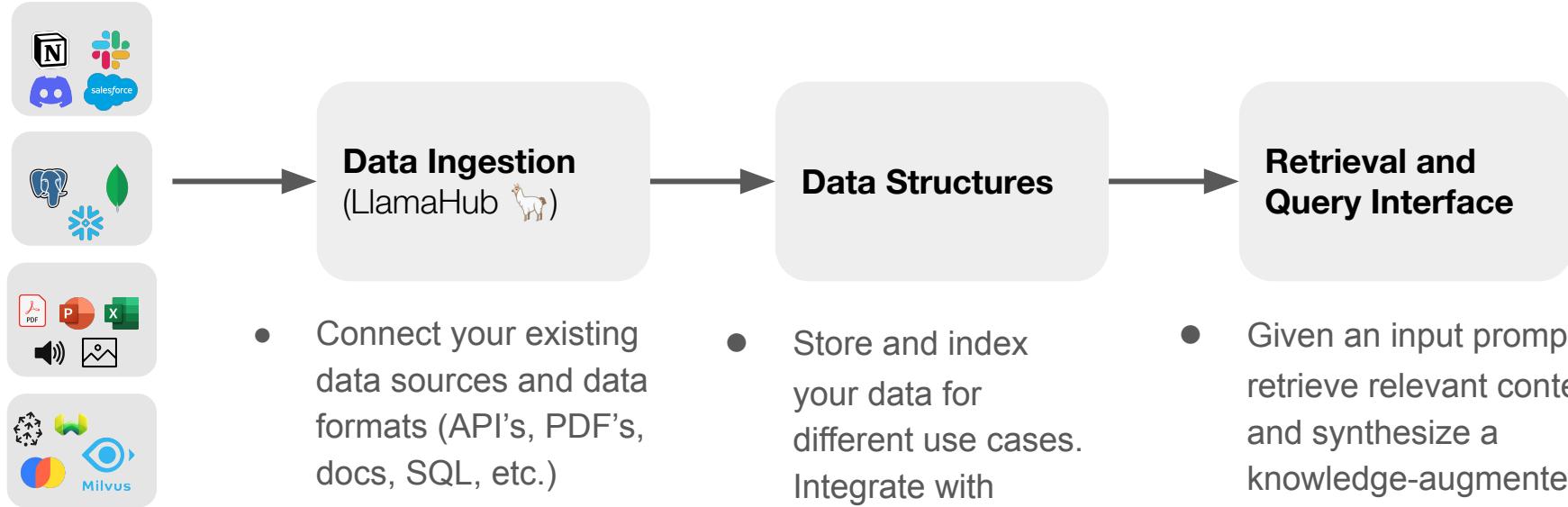
- How to retrieve the right context for the prompt?
- How to deal with long context?
- How to deal with source data that is potentially very large? (GB's, TB's)
- How to tradeoff between:
 - Performance
 - Latency
 - Cost

Llamaindex

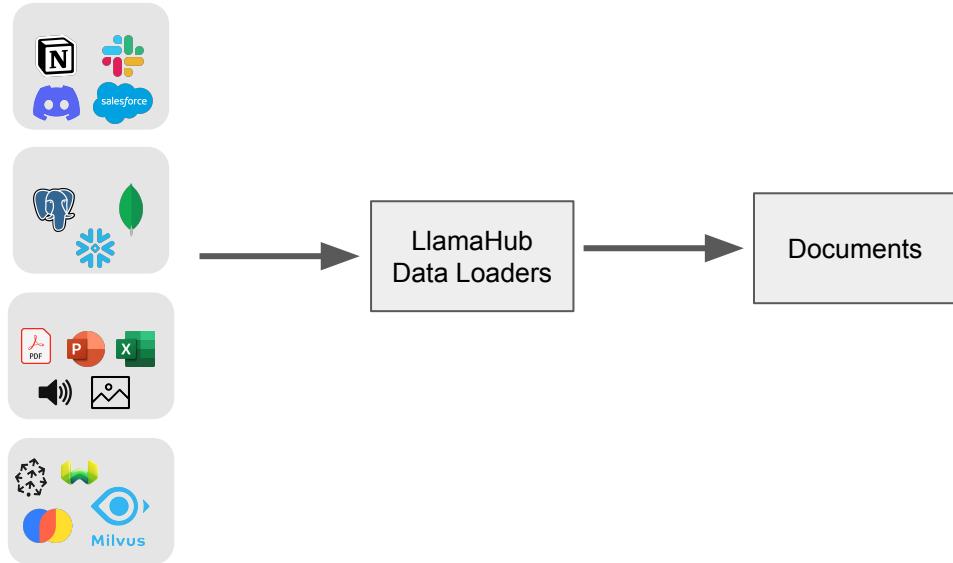


Llamaindex: A data framework for LLM applications

- Data Management and Query Engine for your LLM application
- Offers components across the data lifecycle: ingest, index, and query over data



LlamaIndex: First Abstractions: Data Ingestion (LlamaHub)



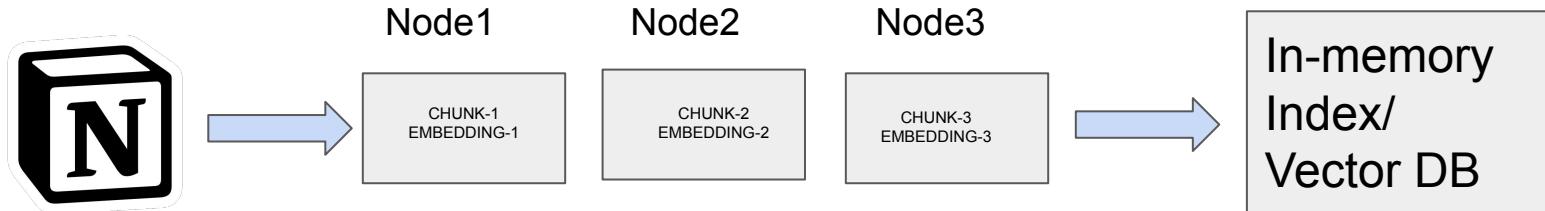
The screenshot shows the Llama Hub interface with a search bar at the top. Below it is a section titled "Data Loaders" with a dropdown menu. The main area displays a grid of data loader cards, each with a name, author, and a "Loader" button. Some cards also show a star rating and a timestamp.

Name	Author	Last Updated	Rating	Action
airtable	simyja	1 month ago	1	Loader
apify/actor	drobnikj	1 month ago	1	Loader
apify/dataset	drobnikj	1 month ago	1	Loader
asana	daveeey	1 month ago	1	Loader
azstorage_blob	rivms	1 month ago	1	Loader
bilibili	alexzhangji	1 month ago	1	Loader
boarddocs	dweekly	1 month ago	1	Loader
chatgpt_plugin	jerryjliu	2 months ago	1	Loader
chroma	atroy	22 days ago	1	Loader
confluence	zywilliamli	1 month ago	1	Loader
couchdb	technosophy	1 month ago	1	Loader
dad_jokes	sidu	1 month ago	1	Loader
database	kevingqz	1 month ago	40	Loader
deeplake	adolikhhan	2 months ago	1	Loader
discord	jerryjliu	1 month ago	24	Loader
docugami	tjaffri	28 days ago	0	Loader
elasticsearch	jaymiller	1 month ago	1	Loader
faiss	jerryjliu	2 months ago	2	Loader

100+ Data Loaders in LlamaHub

Llamaindex: First Abstractions - Indexing

- Indexing:
 - Document (text) is split into multiple chunks and embedding for each chunk is created.
 - Chunk + Embedding is abstracted as Node.
 - All nodes are stored in in-memory index or Vector DB.
 - *Document from different sources are broken down as per settings*

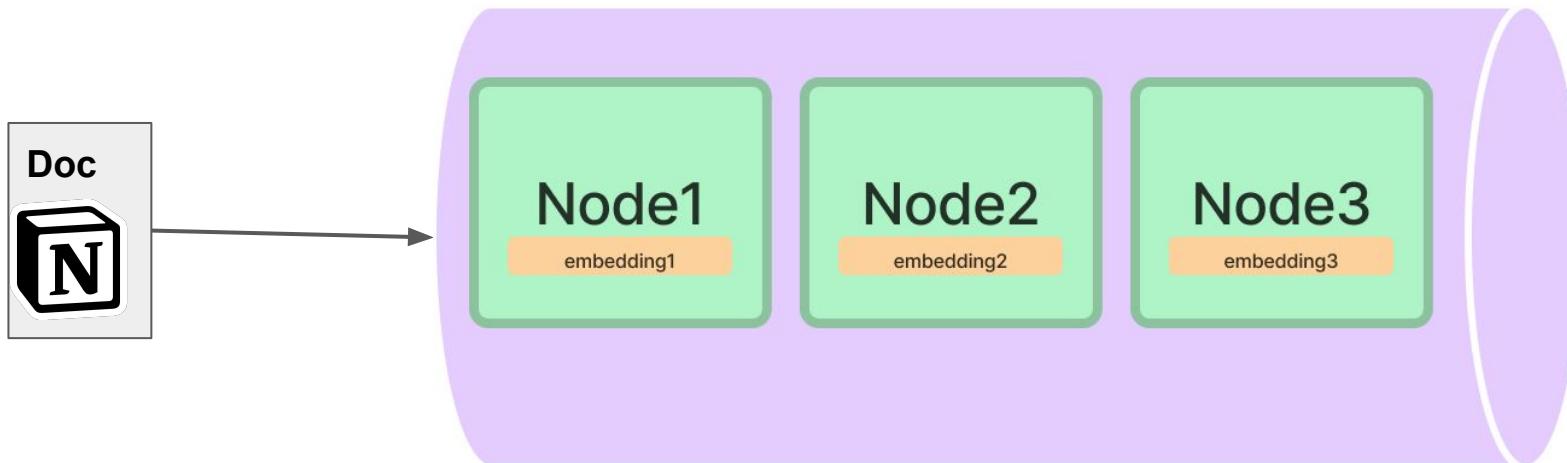


- Works with different vector DB's such as Pinecone, Weaviate, Qdrant.

Vector Store Index

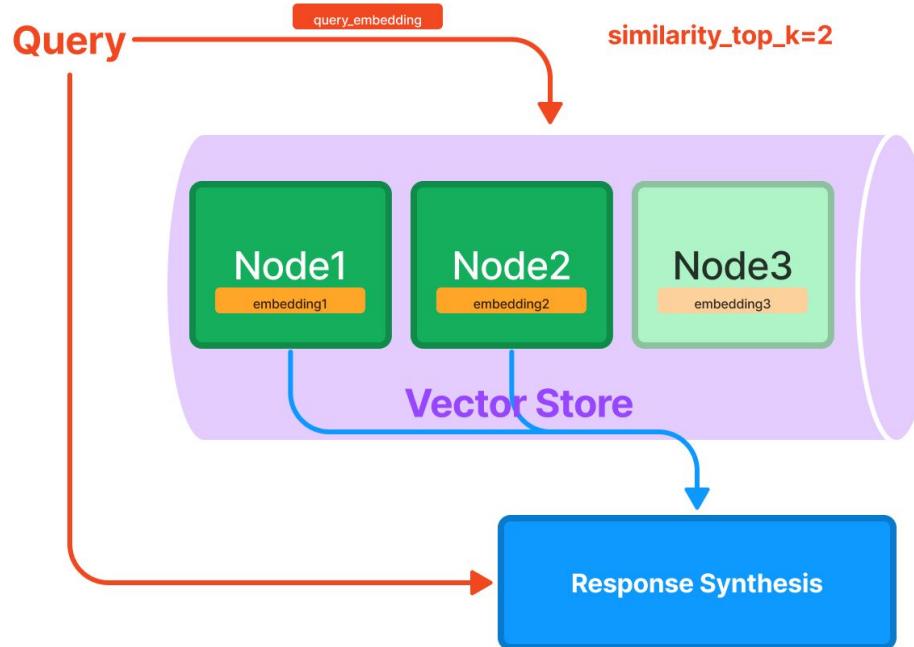
Raw Documents

Stored as Nodes in a vector store
Each Node is indexed with an embedding

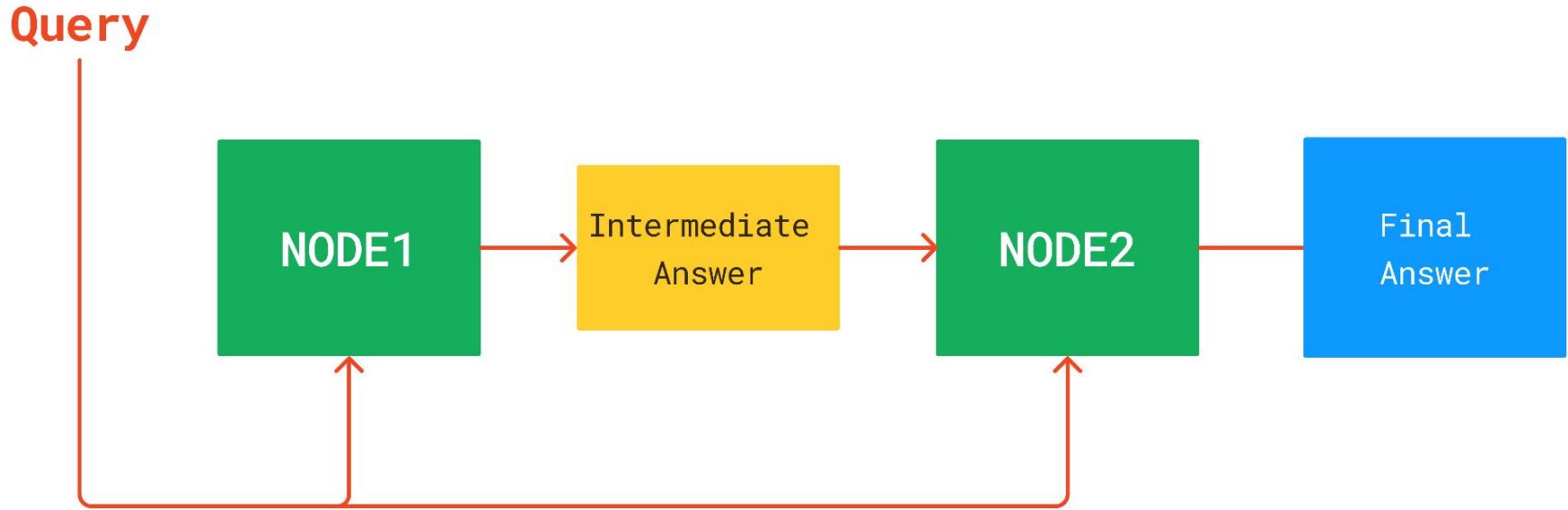


Llamaindex: First Abstractions - Retrieval

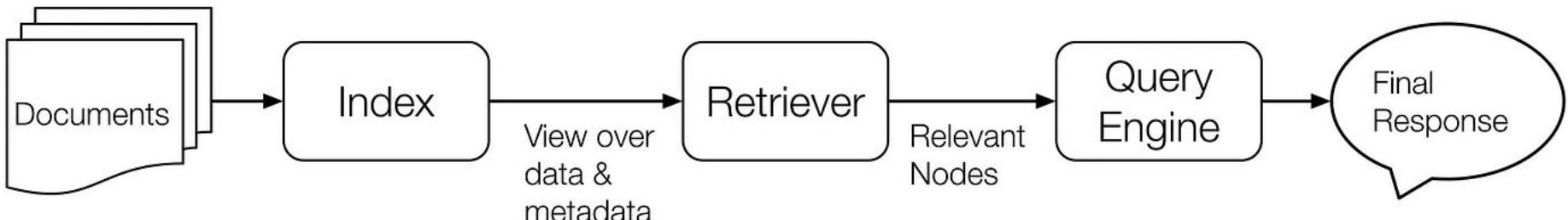
- When user asks a query.
 - It retrieves the nodes and ranks them based on embedding similarity between query and nodes.
 - Post-process the retrieved nodes if needed.
 - Directs the retrieved nodes to Response Synthesis module.



LlamalIndex: First Abstractions - Response Synthesis



Data Indices + Query Interface



Your **source documents** are stored in a data collection

In-memory,
MongoDB

Our **data indices** help to provide a view of your raw data

Vectors, keyword lookups,
summaries

A **retriever** helps to retrieve relevant documents for your query

A **query engine** manages retrieval and synthesis given the query.

Documents



10-K filing with the SEC provides an annual overview of the company's financial performance, operations, and risk factors.

Semantic Search

```
from llama_index import VectorStoreIndex, SimpleDirectoryReader
documents = SimpleDirectoryReader('data').load_data()
index = VectorStoreIndex.from_documents(documents)

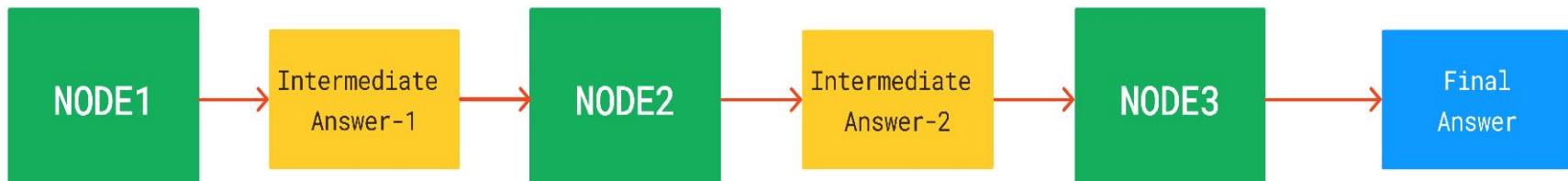
query_engine = index.as_query_engine()
response = query_engine.query(
    "What is the revenue of uber in 2021?"
)
```

Answer

17,455 million

Summarization - SummaryIndex

Query



Summarization

```
from llama_index import SummaryIndex, SimpleDirectoryReader
documents = SimpleDirectoryReader('data').load_data()
index = SummaryIndex.from_documents(documents)

query_engine = index.as_query_engine()

response = query_engine.query("Provide the summary.")
```

Answer

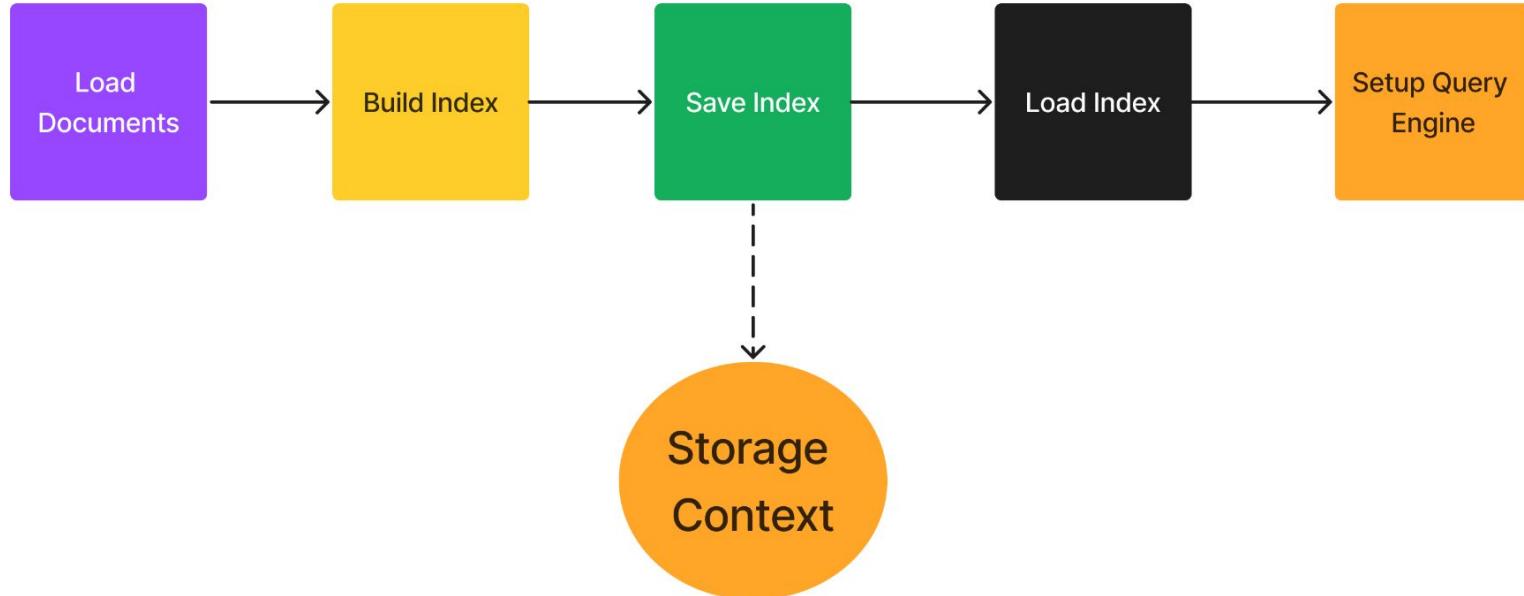
- The author began writing and programming before college, and studied philosophy in college before switching to AI.
- He realized that AI, as practiced at the time, was a hoax and decided to focus on Lisp hacking instead.
- He wrote a book about Lisp hacking and graduated with a PhD in computer science.
-

Building RAG with LlamalIndex

PRACTICAL SESSION - 1

BREAK - 15 minutes

Saving And Loading the index





```
from llama_index import StorageContext, load_index_from_storage
from llama_index.node_parser import SimpleNodeParser

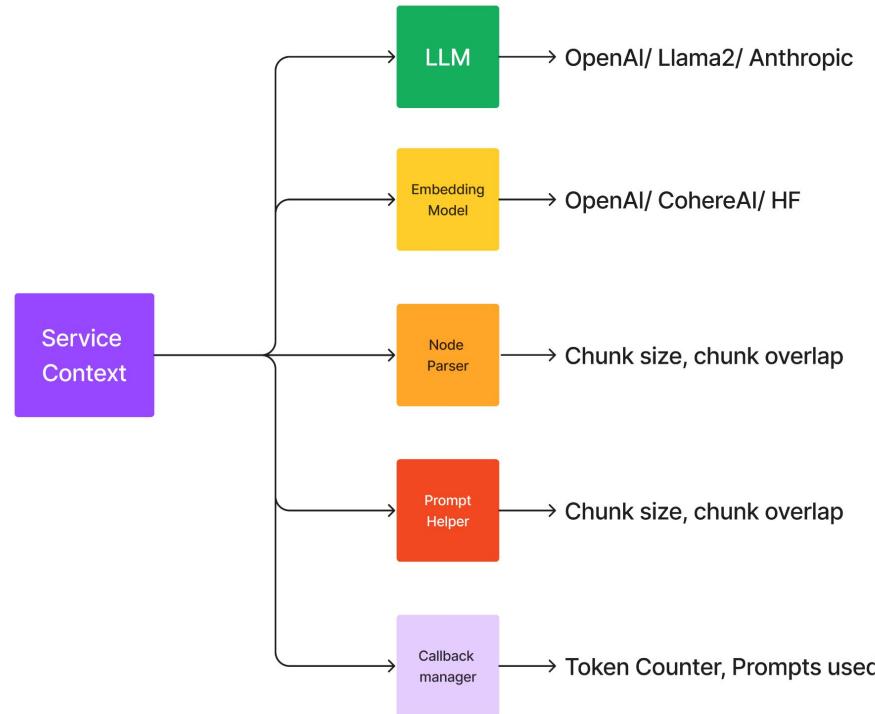
node_parser = SimpleNodeParser.from_defaults(chunk_size=2000, chunk_overlap=100)
nodes = node_parser.get_nodes_from_documents(documents)

storage_context = StorageContext.from_defaults()
index = VectorStoreIndex(nodes, storage_context=storage_context, service_context = service_context)
index.storage_context.persist(persist_dir="storage")
storage_context = StorageContext.from_defaults(persist_dir="storage")

loaded_index = load_index_from_storage(storage_context = storage_context,
service_context=service_context)

query_engine = loaded_index.as_query_engine(similarity_top_k=3)
response = query_engine.query("why did paul graham start YC?")
```

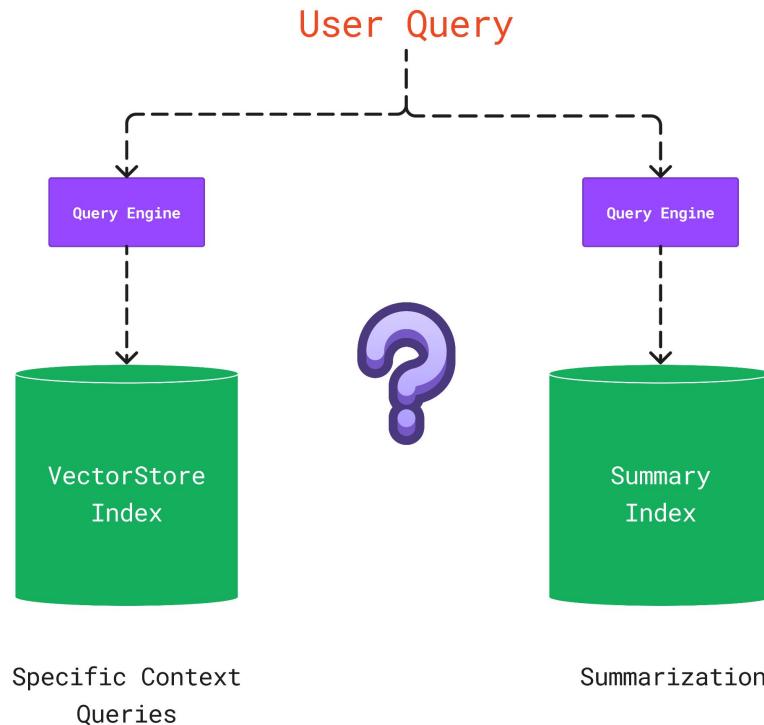
Customizing LLM, Embeddings, Nodes



Customize RAG with Llamaindex

PRACTICAL SESSION - 2

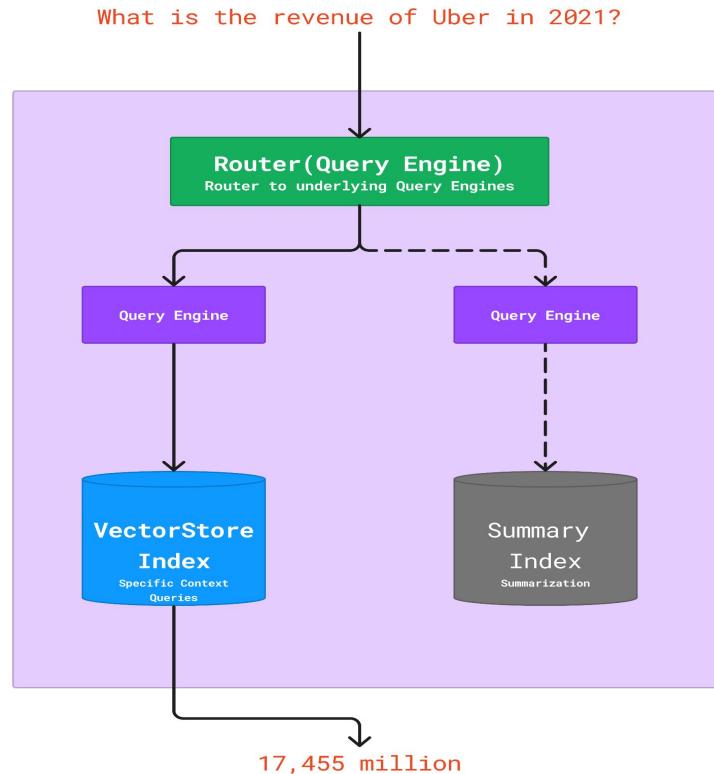
Which Index to query?



Building a Unified Query Interface

Can use a “Router” abstraction to route to different query engines.

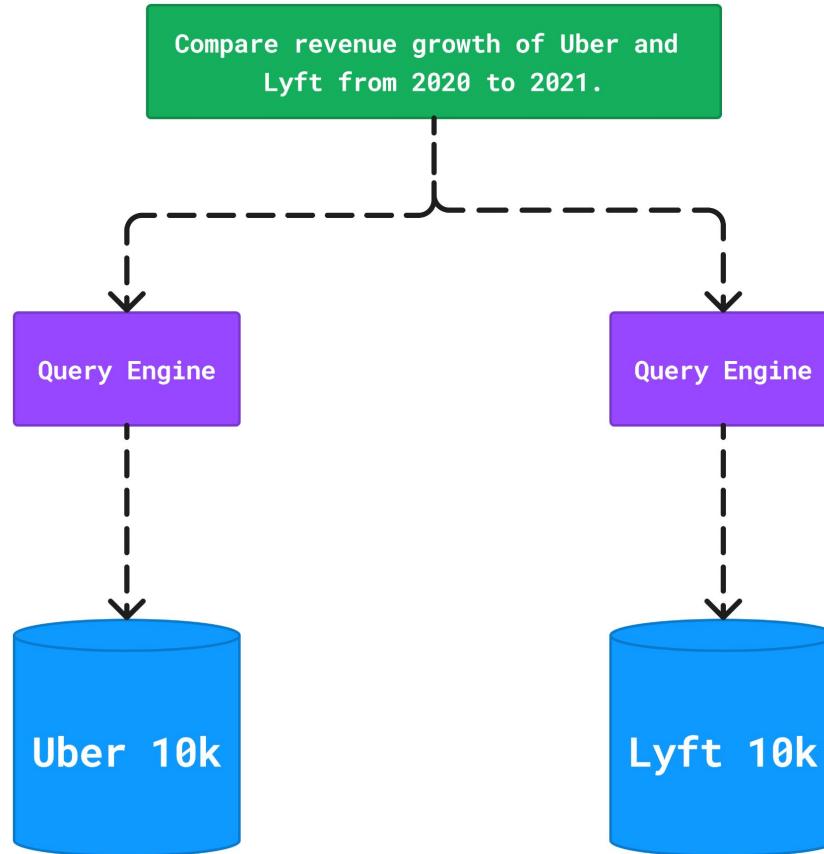
For instance, can do joint semantic search / summarization



Router Query Engine

PRACTICAL SESSION - 3

Document Comparisons

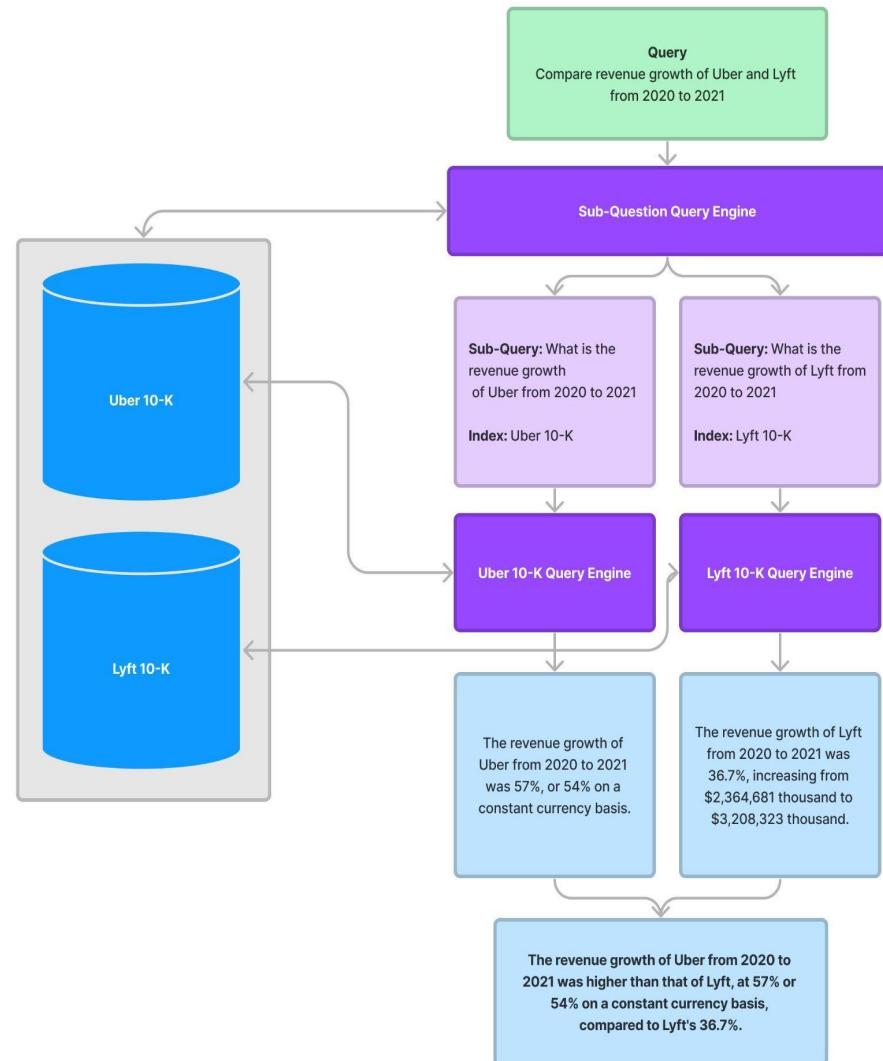


Document Comparisons

Say you want to compare the 2021 10-K filings for Uber and Lyft

Question: “Compare and contrast the customer segments and geographies that grew the fastest.”

Generate a **query plan** over your document sources.



Sub Question Query Engine

PRACTICAL SESSION - 4

BREAK - 15 minutes

Metadata Management

Metadata Filtering

PRACTICAL SESSION-5

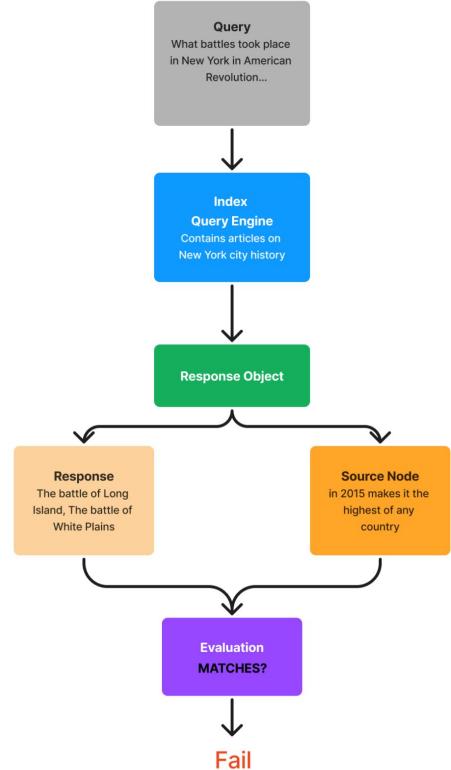
Document Agents with Recursive Retriever

PRACTICAL SESSION - 7

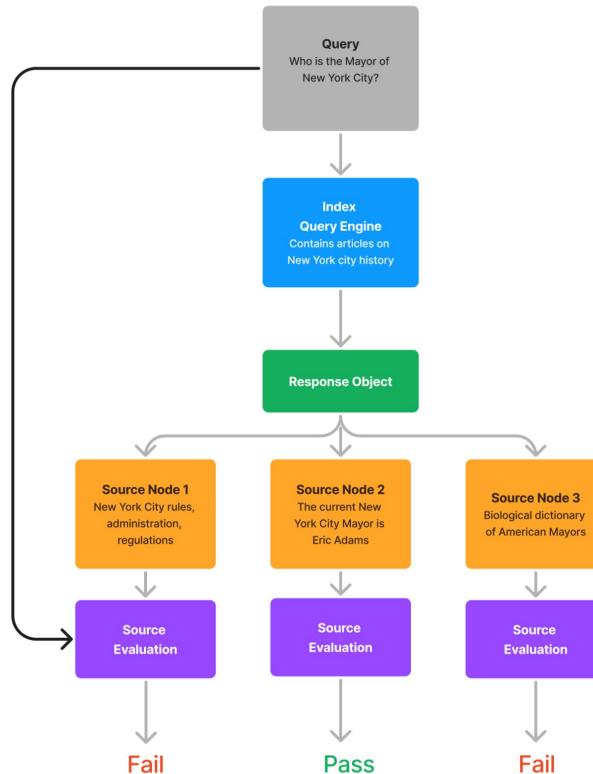
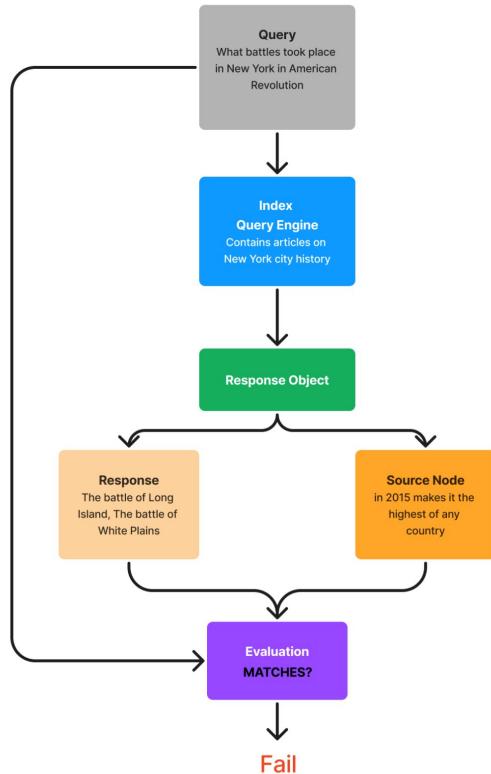
Evaluation

- **Response Evaluation.**
 - **FaithFullNess Evaluator** - to measure if the response is hallucinated.
 - **Relevancy Evaluator** - to measure if the response + source nodes match the query.
 - **Correctness Evaluator** - to evaluate the relevance and correctness of a generated answer against a reference answer.
 - **Guideline Evaluator** - to evaluate a question answer system given user specified guidelines.
 - GUIDELINES = [
 - "The response should fully answer the query.",
 - "The response should avoid being vague or ambiguous.",
 - "The response should be specific and use statistics or numbers when possible.",

FaithFullness Evaluator

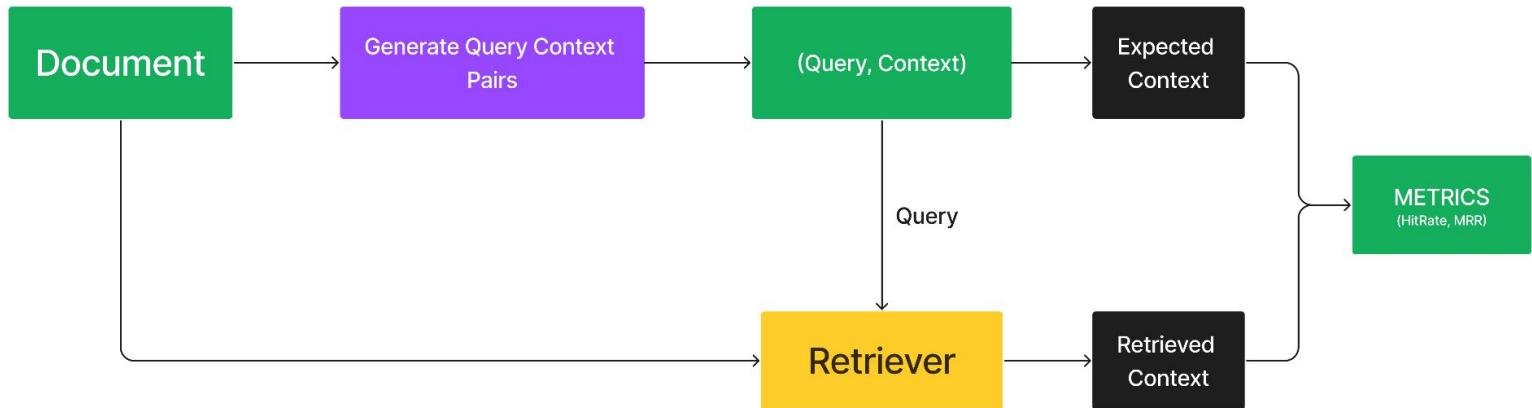


Relevancy Evaluator



Evaluation

- **Retrieval Evaluation.**
 - Hit Rate.
 - MRR



Evaluation guide:

https://gpt-index.readthedocs.io/en/stable/core_modules/supporting_modules/evaluation/root.html

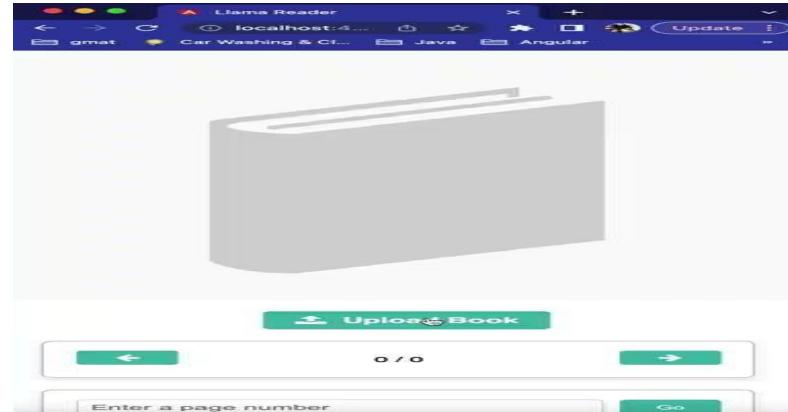
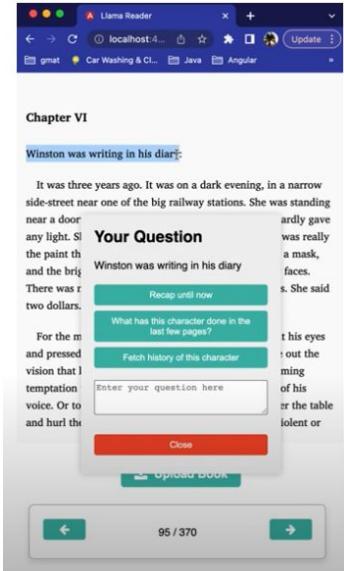
Evaluation

PRACTICAL SESSION - 6

Fine-tuning Embedding Models

Applications Using LlamaIndex

LlamaReaders



Albus

Thread albus-internal

ravi theja 2 minutes ago
@albusdev How many paid leaves are employees entitled to annually?

1 reply

albusdev APP 2 minutes ago

Internal sources ChatGPT

In the context of Springrole India Pvt. Ltd., employees are entitled to 18 days of paid leave annually, in conformity with The Karnataka Shops and Commercial Establishments Act, 1961. This is in addition to 8 fixed paid holidays on national holidays and festivals, as well as 4 "floater leaves" that employees can choose at their discretion. Employees are also entitled to Personal Time Off (PTO) on a prorated basis, depending on the time they join or exit the organization. PTO can be used for various reasons, including mental wellness, body pains, headaches, menstrual pains, or even if an employee doesn't feel like working.

In case of longer illnesses that require extended recovery time, employees can take a maximum of 7 days of leave in case of hospitalization, which includes weekends. After exhausting PTO and Paid Leaves, the company's management may decide on additional leave in exceptional circumstances.

All accumulated PTO leaves will lapse at the end of the year, with no carry-forward option.

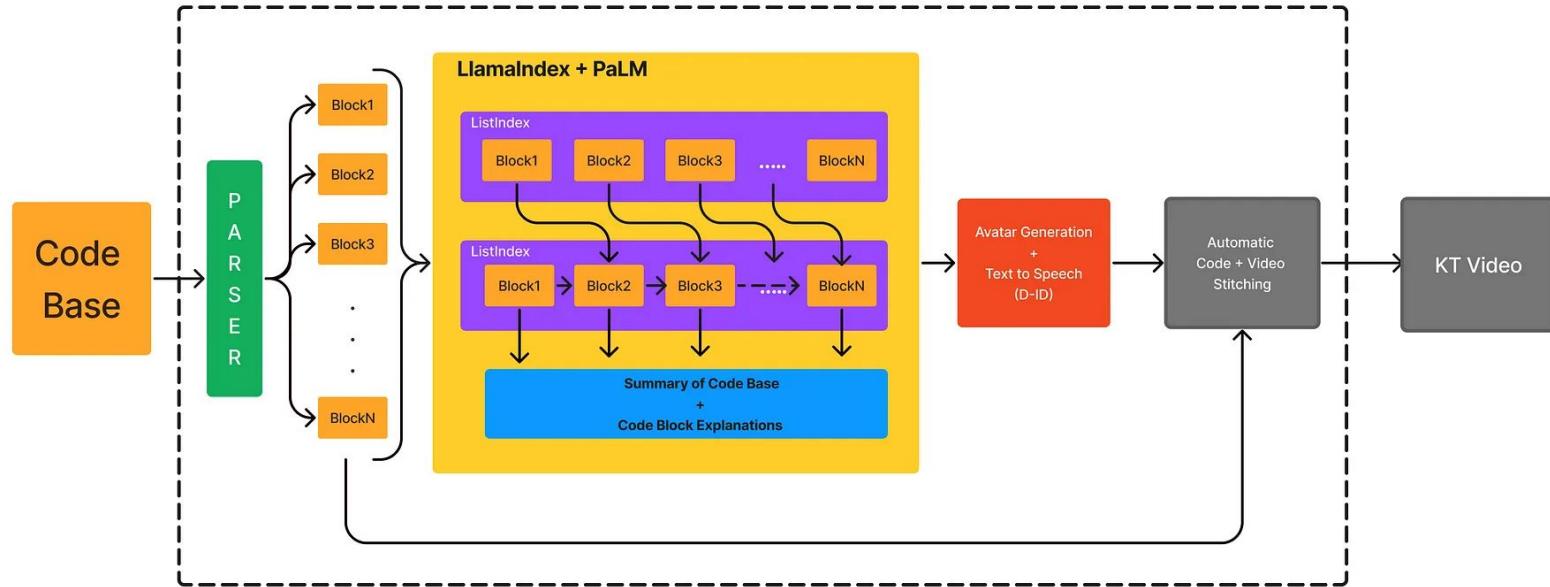
Additionally, the company offers 5 days of Wedding/Marriage Leave with a 30-day notice period and no salary deduction. Female employees are also entitled to Maternity Leave in accordance with the Maternity Benefit Act, 1961, which allows for a maximum of 26 weeks of leave (12 weeks for those expecting their third child). The leave can be split into pre-natal and post-delivery periods.

Please note that these policies may vary depending on the company and jurisdiction.

See less

Demo

Automatic KT Generator For Code Bases



[Video](#)

[Blog Post](#)

BlogPost/ Project

- [BlogPost/ Project Submission](#)

Guides

- [Documentation Guide.](#)
- [Contributions Guide.](#)
- [Discord.](#)
- [Twitter.](#)
- [Linkedin.](#)

Thanks!

[Linkedin](#) to reach out.

[BlogPost/ Project Submission](#)

Win T-Shirts by Participating In Quiz

