



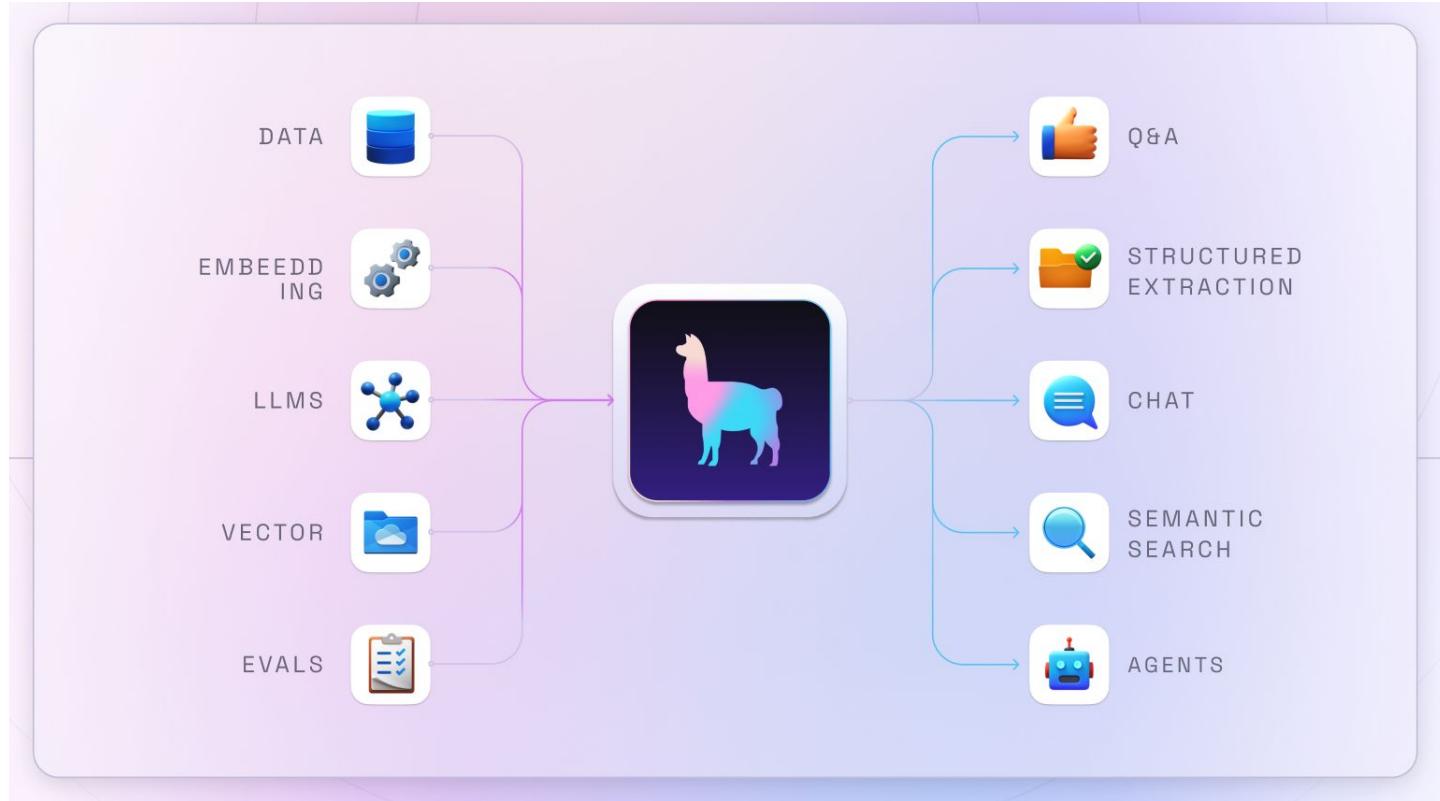
An introduction to RAG in 2024

Pierre-Loic Doulcet, Founding AI engineer, LlamalIndex

@EPFL Lauzhack - May 8th 2024

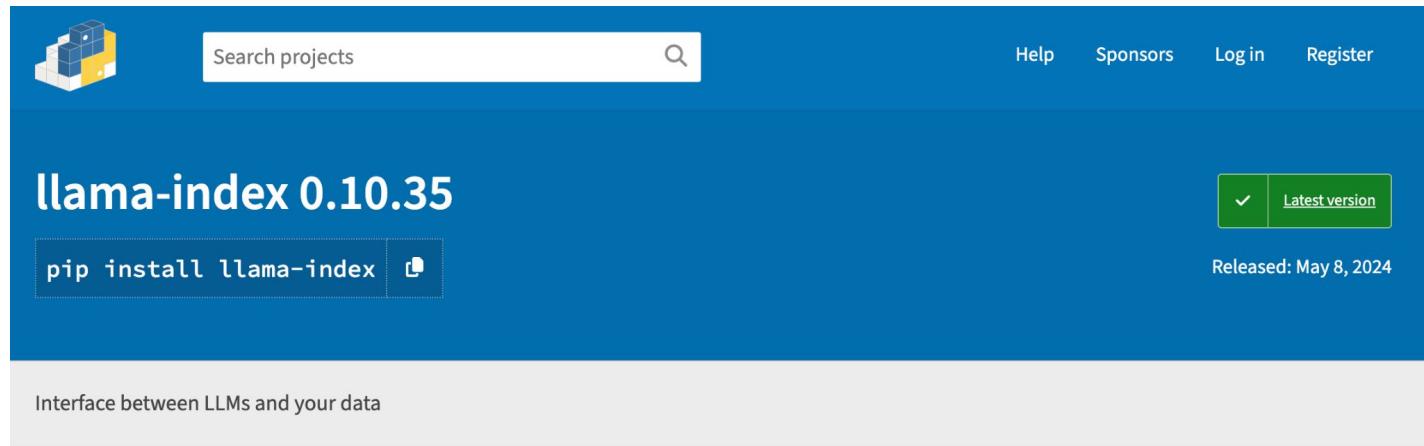


Llamaindex: Context Augmentation for your LLM app





Llamaindex python framework



The screenshot shows the PyPI project page for "llama-index 0.10.35". The header includes a search bar, navigation links for Help, Sponsors, Log in, and Register, and a logo icon. The main title is "llama-index 0.10.35" with a green "Latest version" button. Below the title is a pip install link: "pip install llama-index" with a copy icon. To the right is the release date: "Released: May 8, 2024". A sub-header below the title reads "Interface between LLMs and your data".

Navigation

Project description

Release history

Download files

Project description



downloads 958k/month contributors 481 chat 2212 online P Phorm Ask AI

Llamaindex (GPT Index) is a data framework for your LLM application. Building with Llamaindex typically involves working with Llamaindex core and a chosen set of integrations (or plugins). There are two ways to start building with Llamaindex in Python:

1. Starter: [llama-index](https://pypi.org/project/llama-index/) (<https://pypi.org/project/llama-index/>). A starter Python package that includes core Llamaindex as well as a selection of integrations.

Verified details

These details have been verified by PyPI



Llamaindex typescript framework

Llamaindex 

0.3.8 • Public • Published 19 hours ago

 Readme

 Code 

 38 Dependencies

 16 Dependents

 113 Versions

Llamaindex.TS

 v0.3.8  MIT  61k/month  2211 online

Llamaindex is a data framework for your LLM application.

Use your own data with large language models (LLMs, OpenAI ChatGPT and others) in Typescript and Javascript.

Documentation: <https://ts.llamaindex.ai/>

Install

`> npm i llamaindex`



Repository

 github.com/run-llama/LlamaindexTS

Homepage

 github.com/run-llama/LlamaindexTS#r...



LlamaHub.ai

**LlamaHub**
Powered by LlamaIndex

[Github](#)

Our integrations include utilities such as Data Loaders, Agent Tools, Llama Packs, and Llama Datasets. We make it extremely easy to connect large language models to a large variety of knowledge & data sources. Use these utilities with a framework of your choice such as LlamaIndex, LangChain, and more. [Learn More](#)

CSVReader

[Loaders](#) llama-index ⭐ 52 •  491578 • 19 hours ago

DocxReader

 Verified[Loaders](#) thejessezhang ⭐ 52 •  491578 • 19 hours ago

EpubReader

[Loaders](#) haowjy ⭐ 52 •  491578 • 19 hours ago

FlatReader

[Loaders](#) llama-index ⭐ 52 •  491578 • 19 hours ago

HTMLTagReader

[Loaders](#) llama-index ⭐ 52 •  491578 • 19 hours ago

HWPReader

[Loaders](#) sangwongenip ⭐ 52 •  491578 • 19 hours ago

IPYNBReader

[Loaders](#) FarisHijazi ⭐ 52 •  491578 • 19 hours ago

ImageCaptionReader

[Loaders](#) FarisHijazi ⭐ 52 •  491578 • 19 hours ago

ImageReader

[Loaders](#) ravi03071991 ⭐ 52 •  491578 • 19 hours ago

ImageTabularChartReader

[Loaders](#) jon-chuang ⭐ 52 •  491578 • 19 hours ago

ImageVisionLLMReader

[Loaders](#) FarisHijazi ⭐ 52 •  491578 • 19 hours ago

MarkdownReader

[Loaders](#) hursh-desai ⭐ 52 •  491578 • 19 hours ago



Create Llama - jump start your llm app development.

```
>> npm create llama@latest
Need to install the following packages:
| create-llama@latest
Ok to proceed? (y) y
✓ What is your project named? ... my-app
✓ Which template would you like to use? > Chat
✓ Which framework would you like to use? > NextJS
✓ Would you like to set up observability? > No
✓ Please provide your OpenAI API key (leave blank to skip):
✓ Which data source would you like to use? > Use an example
✓ Would you like to add another data source? > No
✓ Would you like to use LlamaParse (improved parser for RAG)
✓ Would you like to use a vector database? > No, just store
? How would you like to proceed? > - Use arrow-keys. Return
    Just generate code (~1 sec)
> Start in VSCode (~1 sec)
Generate code and install dependencies (~2 min)
Generate code, install dependencies, and run the app (~2
```

Get started by editing `app/page.tsx`

Built by LlamaIndex

Length: Not more than 11-1/2 inches
Height: Not more than 6-1/8 inches
Thickness: Not more than 1/4 inch
Weight Standard:
Not more than 3.5 ounces (For First-Class Mail letter-size pieces over 3.5 ounces, flat-size prices apply)
Shape Standards:
Rectangular shape with four square corners and parallel opposite sides
Card-type mailpieces made of cardstock may have finished corners not exceeding a radius of 0.125 inch (1/8 inch)
Nonmachinable Criteria: A letter-size piece is considered nonmachinable if it has one or more of the following characteristics:
Aspect ratio (length divided by height) of less than 1.3 or more than 2.5
Polybagged, polywrapped, enclosed in any non-paper material, or has an exterior surface made of non-paper material
Clasps, strings, buttons, or similar closure devices
Contains items such as pens, pencils, keys, or coins that cause uneven thickness
Too rigid (does not bend easily)
Less than 0.009 inches thick if the mailpiece is more than 6 inches long or 4-1/4 inches high ↗
Delivery address parallel to the shorter dimension of the mailpiece
Not prepared according to specific requirements for self-mailers or booklets
These are the general standards for letters as outlined in the Domestic Mail Manual.

Type a message

Regenerate

Send message



secInsights.ai : Our advanced RAG Demo

Built by Llamaindex 

Empower your organization's Business Intelligence with **SEC Insights**

Effortlessly analyze multifaceted financial documents such as 10-Ks and 10-Qs.

 Open-Sourced on Github

Start your conversation by selecting the documents you want to explore

 Search by company ticker or name ⌘K  Select Document Type  Select Year  Add
Shift + Enter to add to list

 Use the document selector above to start adding documents

Add up to 10 docs to  start your conversation →



LlamaCloud - Managed RAG SaaS

LlamaParse - Doc to structured data Parser

LlamaIndex

jerryjliu98@gmail.com Sign Out

Project: uber_10q_v5

MAIN NAVIGATION: Parse, Index, USAGE, PDF Parser (0 / 1000 pages per day)

Playground: Welcome to the Playground, where you can test transformations and see their impact.

Select Document(s), Select Transformation(s) (highlighted), Configure RAG, Evaluate RAG

Apply Transformation to your RAG pipeline: Select Transformation(s)

Sentence Splitter: Remove, Configure Transformation

OpenAI Embedding: Remove, Configure Transformation

Before, After, Sentence Splitter, uber_10q_sept_2021.pdf

Node 1

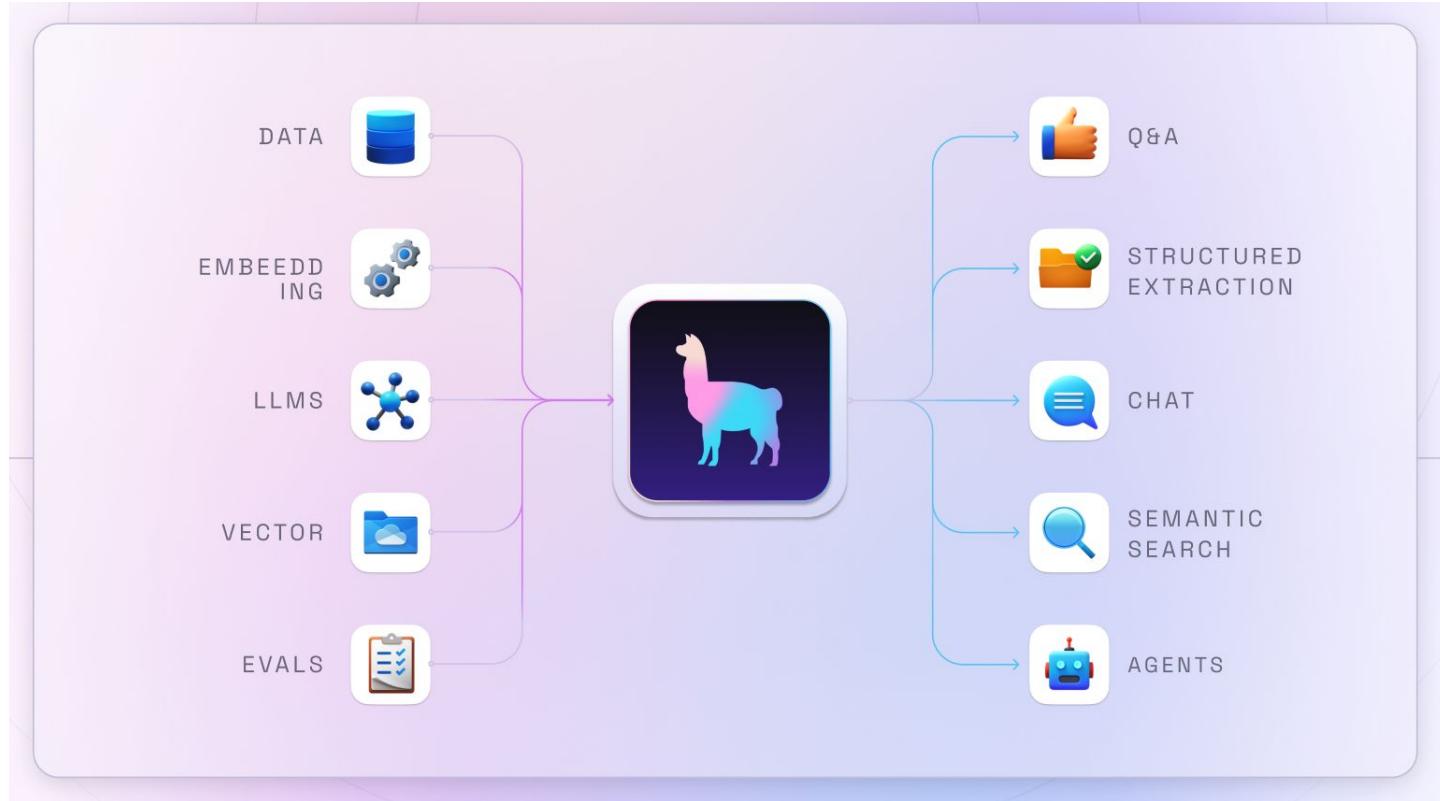
Text:

UNITED STATESSECURITIES AND EXCHANGE COMMISSION Washington, D.C. 20549
FORM 10-Q _____ (Mark One) QUARTERLY REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES
EXCHANGE ACT OF 1934For the quarterly period ended September 30, 2021 or TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE
SECURITIES EXCHANGE ACT OF 1934For the transition period from____ to ____
Commission File Number: 001-38902
UBER TECHNOLOGIES, INC. (Exact name of registrant as specified in its charter)Not Applicable (Former name, former address and former fiscal year, if
changed since last report)_____ Delaware 45-2647441 (State or other jurisdiction of
incorporation or organization)(I.R.S. Employer Identification No.) 1515 3rd Street San Francisco, California 94158 (Address of principal executive offices,
including zip code)

Previous, Deploy, Next



Llamaindex: Context Augmentation for your LLM app





RAG?

(Retrieval Augmented Generation)



Paradigms for inserting knowledge

Retrieval Augmentation - Fix the model, put context into the prompt



Before college the two main things I worked on, outside of school, were writing and programming. I didn't write essays. I wrote what beginning writers were supposed to write then, and probably still are: short stories. My stories were awful. They had hardly any plot, just characters with strong feelings, which I imagined made them deep...



Input Prompt

Here is the context:
Before college the two main things...



Given the context,
answer the following
question:
{query_str}

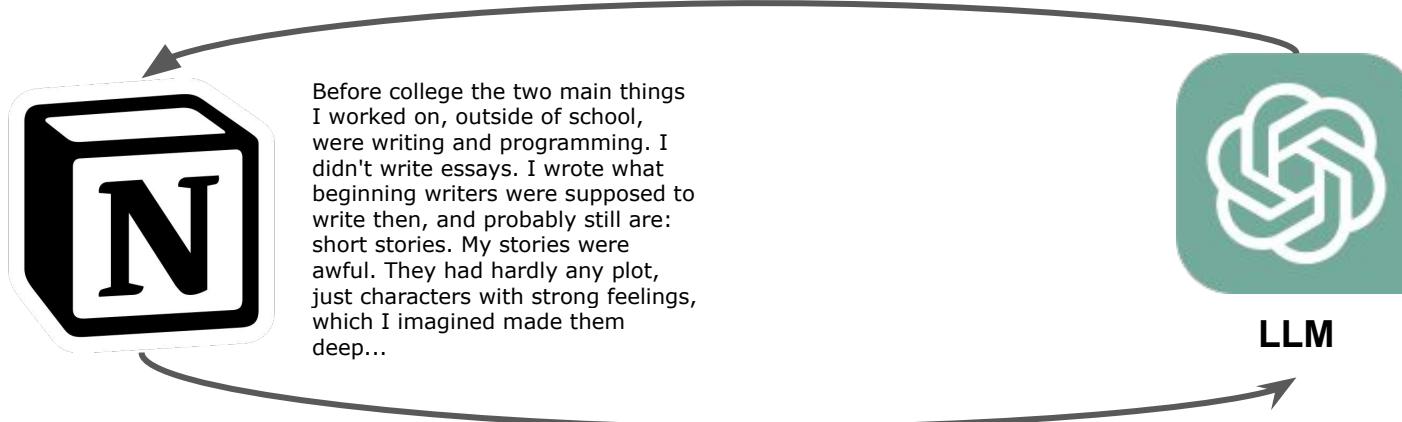


LLM



Paradigms for inserting knowledge

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

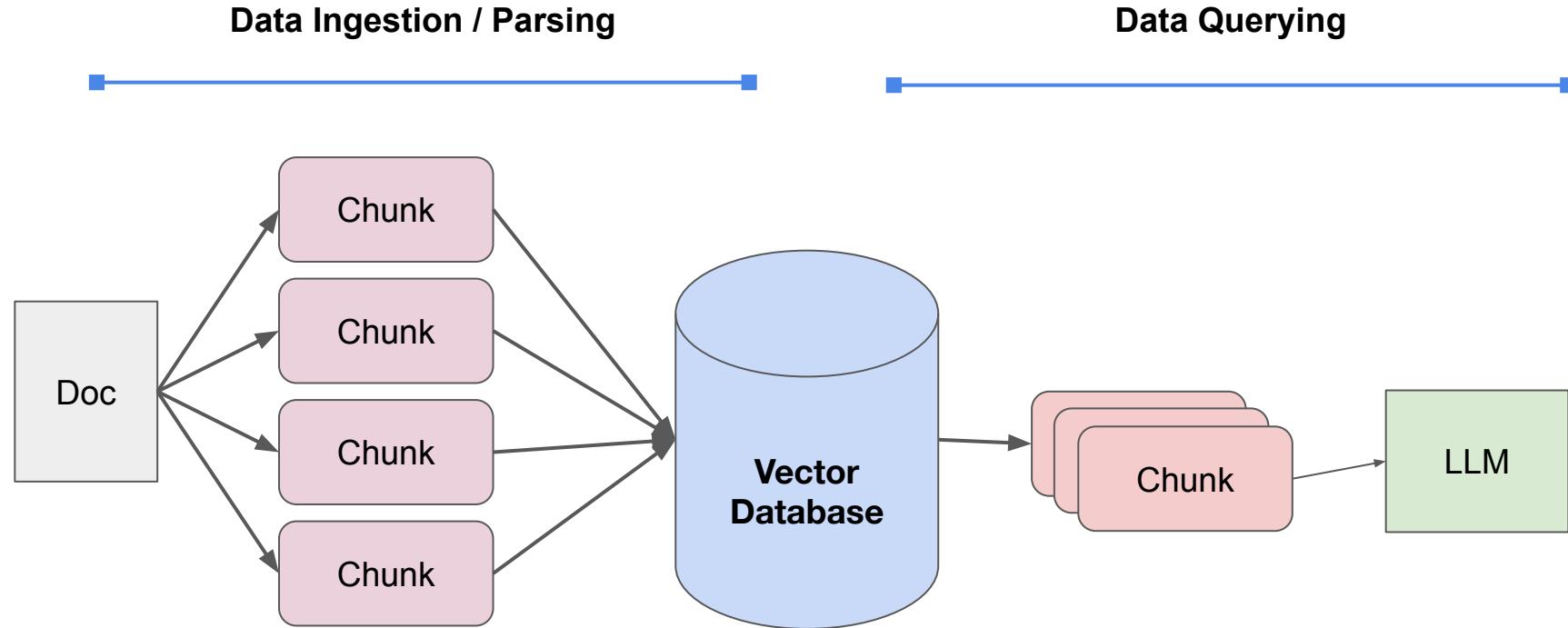




RAG Stack



Current RAG Stack for building a QA System



5 Lines of Code in `Llamaindex!`



Current RAG Stack for building a QA System

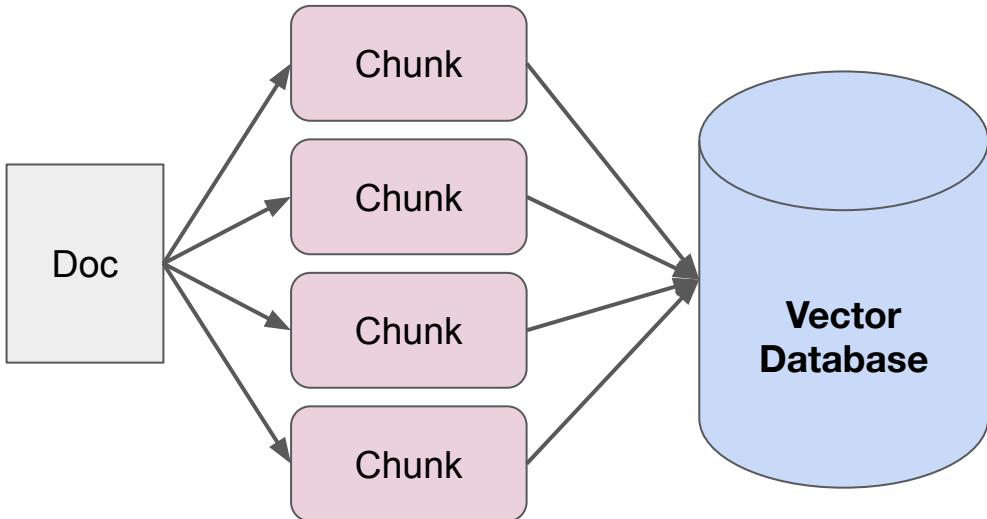
```
1 from llama_index.core import VectorStoreIndex, SimpleDirectoryReader  
2  
3 documents = SimpleDirectoryReader("data").load_data()  
4 index = VectorStoreIndex.from_documents(documents)  
5 query_engine = index.as_query_engine()  
6 response = query_engine.query("What did the author do growing up?")  
7  
8 print(response)
```

5 Lines of Code in Llamalndex!





Current RAG Stack (Data Ingestion/Parsing)

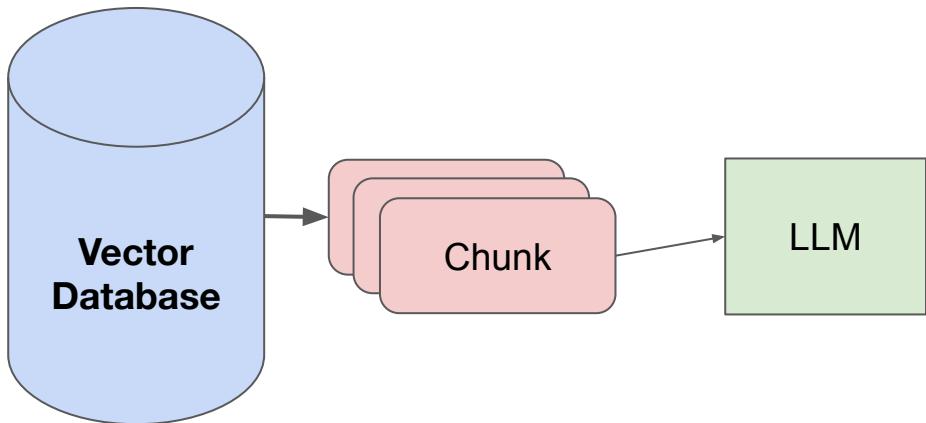


Process:

- Split up document(s) into even chunks.
- Each chunk is a piece of raw text.
- Generate embedding for each chunk (e.g. OpenAI embeddings, sentence_transformer)
- Store each chunk into a vector database



Current RAG Stack (Querying)



Process:

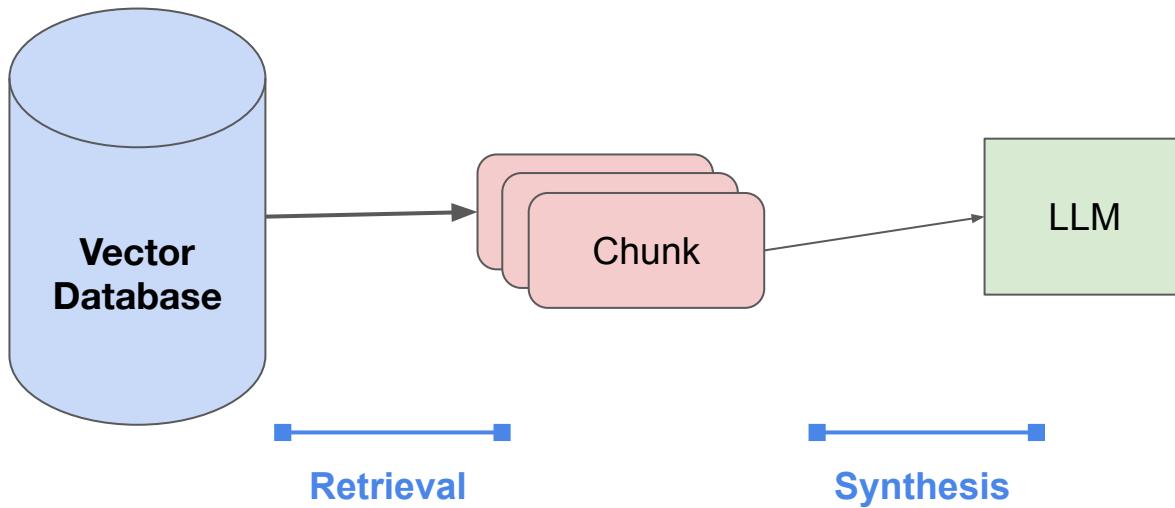
- Find top-k most similar chunks from vector database collection
- Plug into LLM response synthesis module



Current RAG Stack (Querying)

Process:

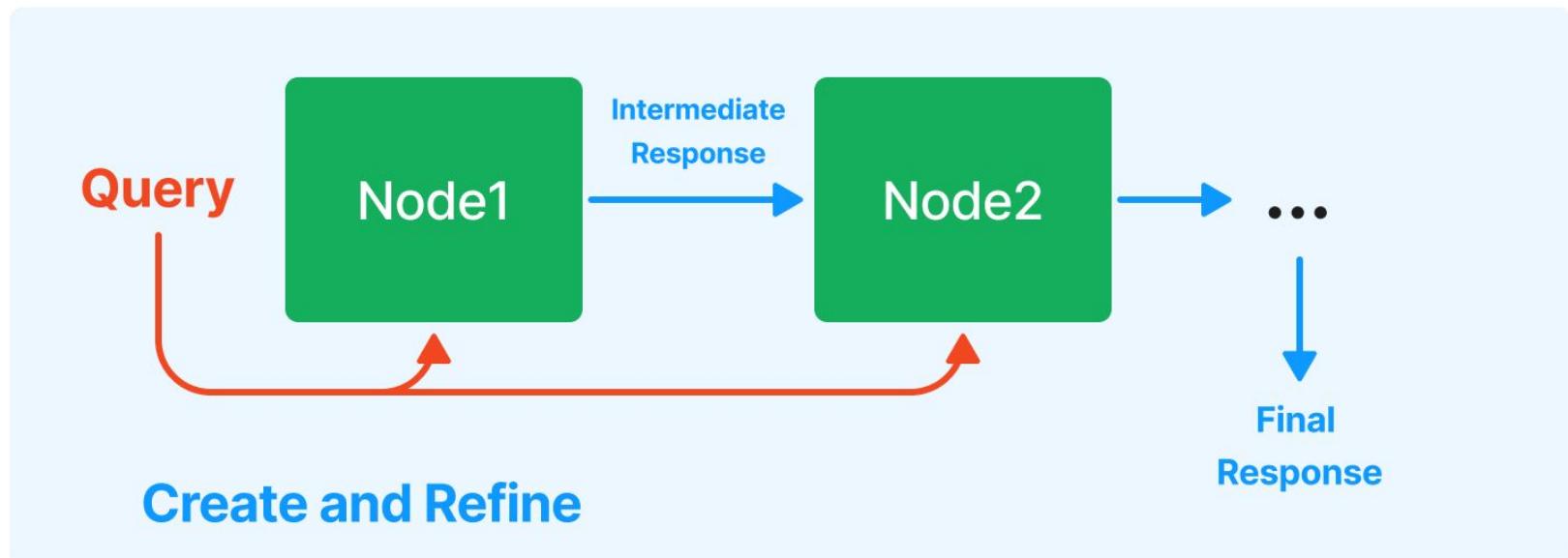
- Find top-k most similar chunks from vector database collection
- Plug into LLM **response synthesis module**





Response Synthesis

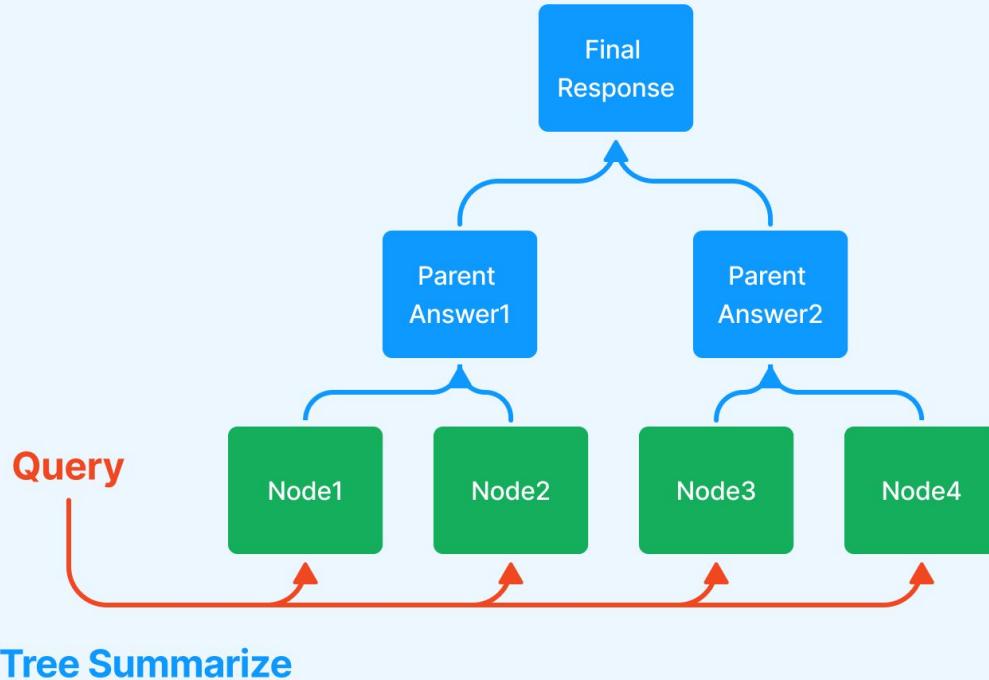
Create and refine





Response Synthesis

Tree Summarize





Quickstart

https://colab.research.google.com/drive/1knQpGJLHj-LTTHqlZhgcjDH5F_nJliY0?usp=sharing



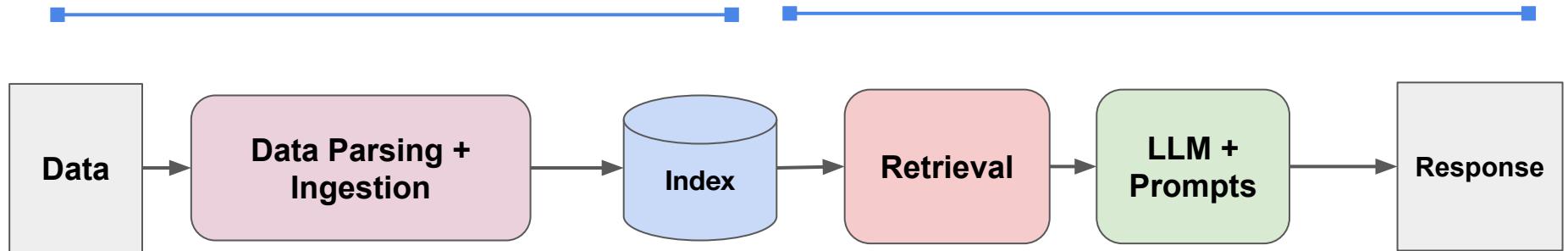


Challenges with “Naive” RAG



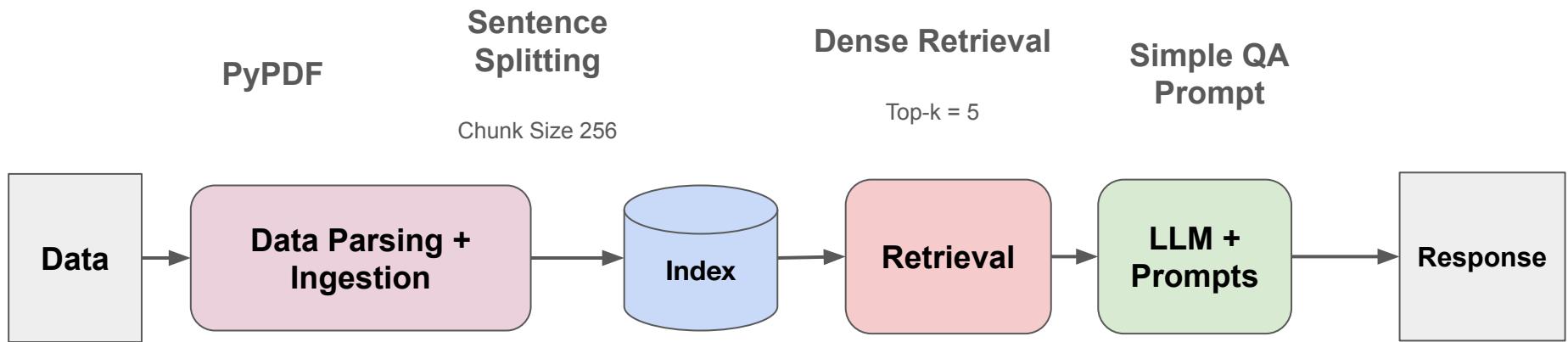
RAG

Data Parsing & Ingestion Data Querying





Naive RAG





Easy to Prototype, Hard to Productionize

Naive RAG approaches tend to work well for **simple** questions over a **simple, small** set of documents.

- “What are the main risk factors for Tesla?” (over Tesla 2021 10K)
- “What did the author do during his time at YC?” (Paul Graham essay)



Easy to Prototype, Hard to Productionize

But productionizing RAG over **more questions** and a **larger set of data** is hard!

Failure Modes:

- Response Quality: Bad Retrieval, Bad Response Generation
- Hard to Improve: Too many parameters to tune
- Systems: Latency, Cost, Security



Easy to Prototype, Hard to Productionize

But productionizing RAG over **more questions** and a **larger set of data** is hard!

Failure Modes:

- **Response Quality:** Bad Retrieval, Bad Response Generation
- **Hard to Improve:** Too many parameters to tune
- **Systems:** Latency, Cost, Security



Challenges with Naive RAG (Response Quality)

- Bad Retrieval
 - **Low Precision:** Not all chunks in retrieved set are relevant
 - Hallucination + Lost in the Middle Problems
 - **Low Recall:** Not all relevant chunks are retrieved.
 - Lacks enough context for LLM to synthesize an answer
 - **Outdated information:** The data is redundant or out of date.



Challenges with Naive RAG (Response Quality)

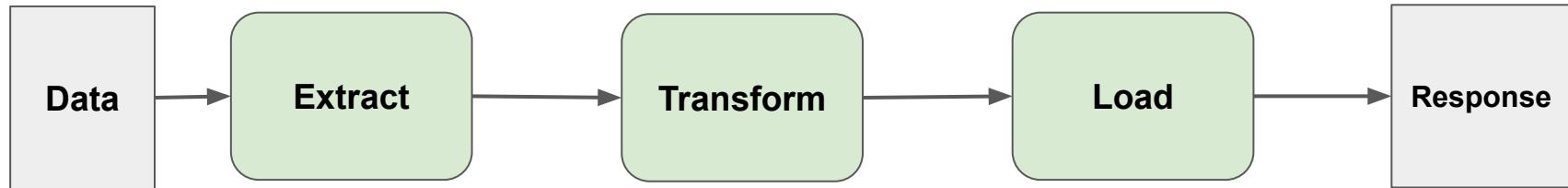
- Bad Retrieval
 - **Low Precision:** Not all chunks in retrieved set are relevant
 - Hallucination + Lost in the Middle Problems
 - **Low Recall:** Not all relevant chunks are retrieved.
 - Lacks enough context for LLM to synthesize an answer
 - **Outdated information:** The data is redundant or out of date.
- Bad Response Generation
 - **Hallucination:** Model makes up an answer that isn't in the context.
 - **Irrelevance:** Model makes up an answer that doesn't answer the question.
 - **Toxicity/Bias:** Model makes up an answer that's harmful/offensive.



Difference with Traditional Software

Traditional software is defined by a set of programmatic rules.

Given an input, you can easily reason about the expected output.



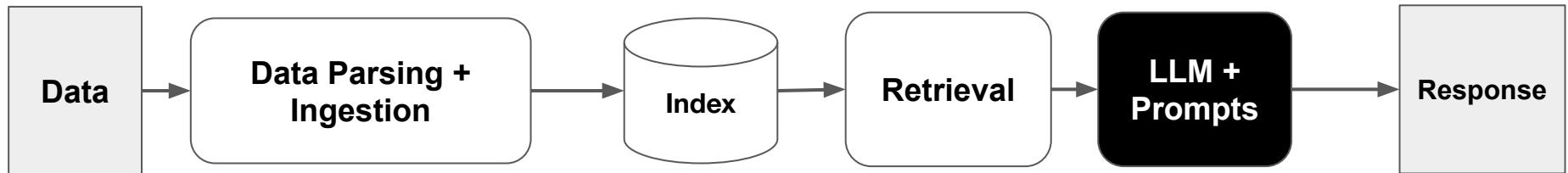
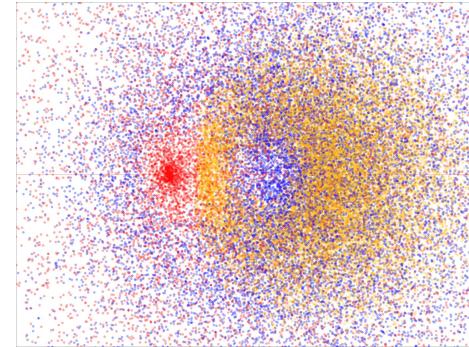


Difference with Traditional Software

AI-powered software is defined by a **black-box set of parameters**.

It is really hard to reason about what the function space looks like.

The model parameters are tuned, the surrounding parameters (prompt templates) are not.

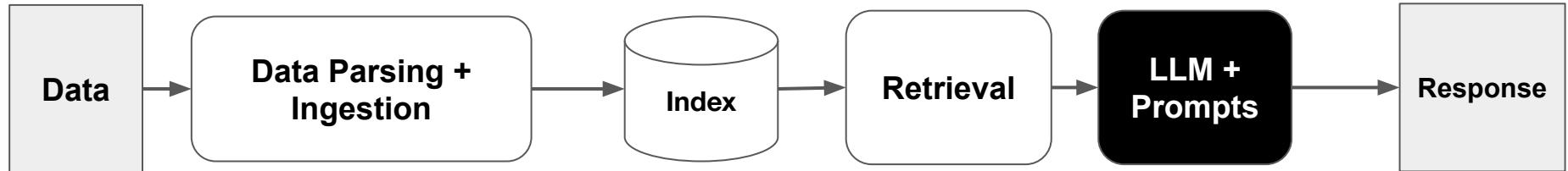




Difference with Traditional Software

If one component of the system is a black-box, all components of the system become black boxes.

The more components, the more parameters you have to tune.

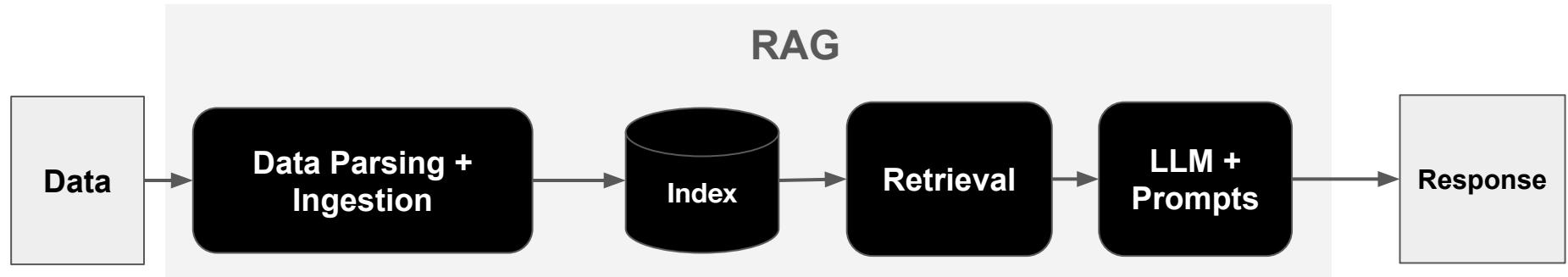




Difference with Traditional Software

If one component of the system is a black-box, all components of the system become black boxes.

Every parameter affects the performance of the end system.



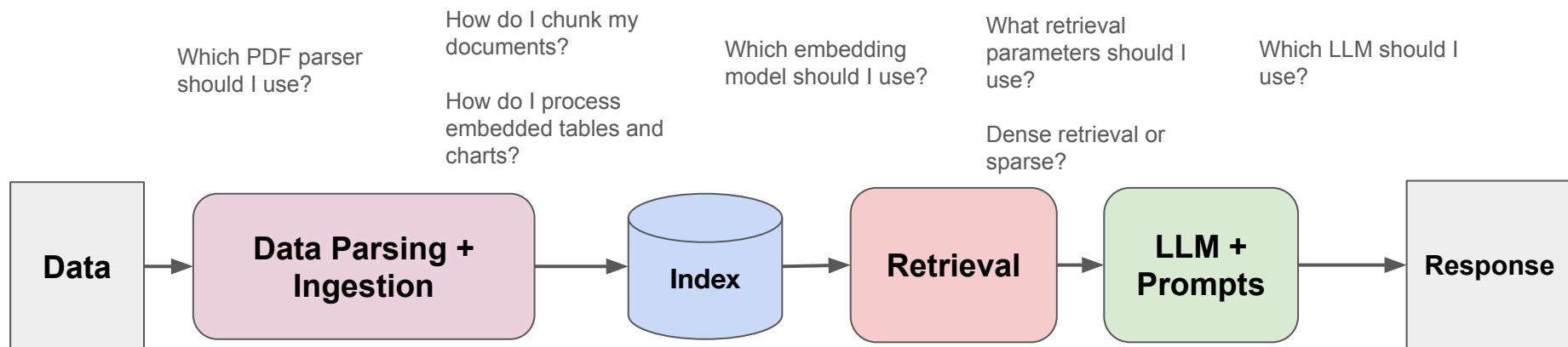


There's Too Many Parameters

Every parameter affects the performance of the entire RAG pipeline.

Which parameters should a user tune?

There's too many options!





Mapping Pain Points to Solutions



Solution

Categorize by pain point, and establish best practices



Solution

Categorize by pain point, and establish best practices

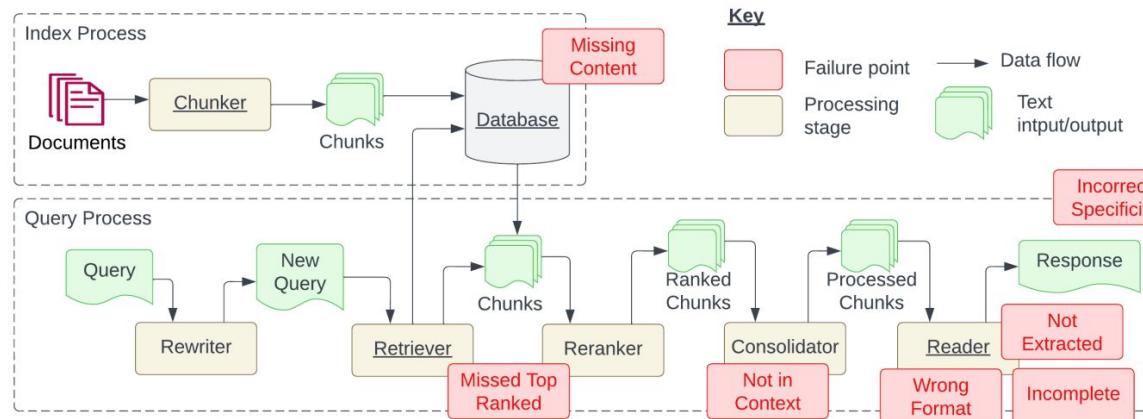


Figure 1: Indexing and Query processes required for creating a Retrieval Augmented Generation (RAG) system. The indexing process is typically done at development time and queries at runtime. Failure points identified in this study are shown in red boxes. All required stages are underlined. Figure expanded from [19].

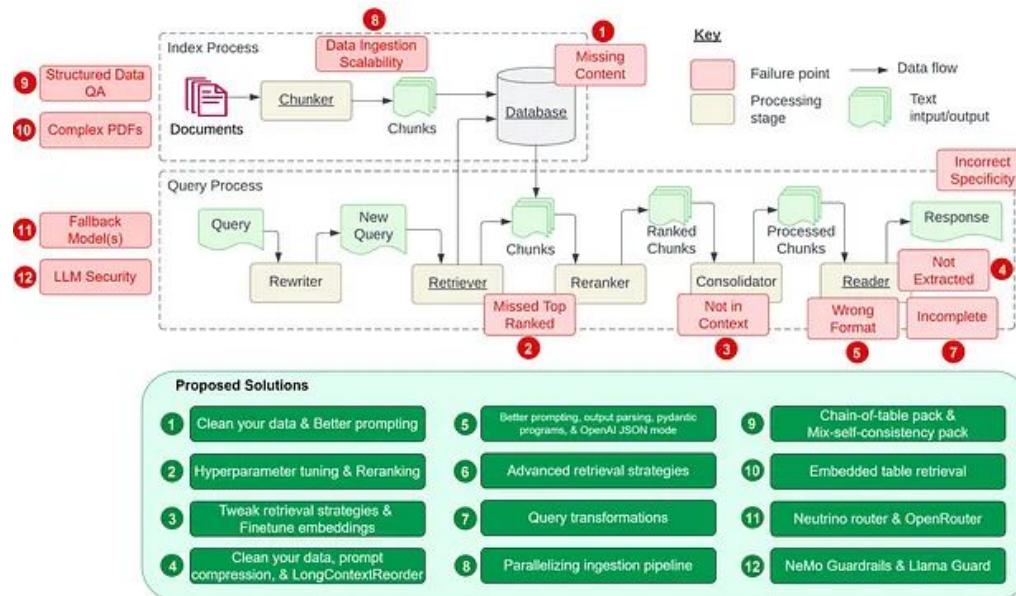
["Seven Failure Points When Engineering a Retrieval Augmented Generation System"](#), Barnett et al.





Solution

Categorize by pain point, and establish best practices



[“12 RAG Pain Points and Proposed Solutions”, by Wengi Glantz](#)





Pain Points

Response Quality Related

1. Context Missing in the Knowledge Base
2. Context Missing in the Initial Retrieval Pass
3. Context Missing After Reranking
4. Context Not Extracted
5. Output is in Wrong Format
6. Output has Incorrect Level of Specificity
7. Output is Incomplete

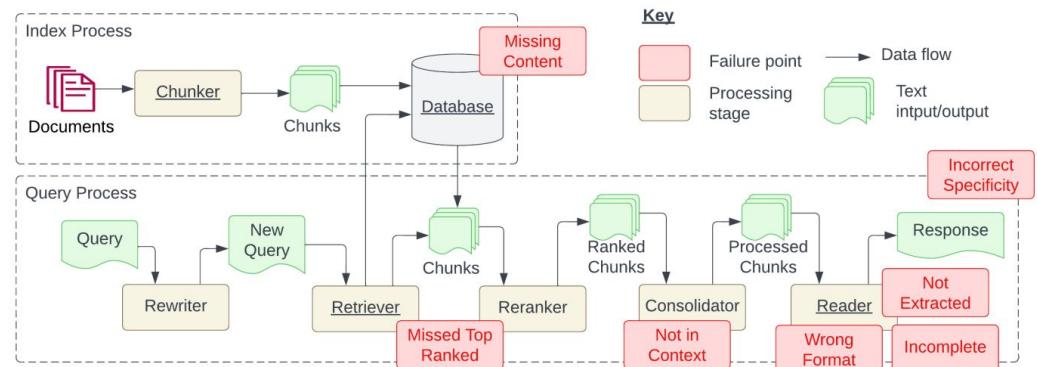


Figure 1: Indexing and Query processes required for creating a Retrieval Augmented Generation (RAG) system. The indexing process is typically done at development time and queries at runtime. Failure points identified in this study are shown in red boxes. All required stages are underlined. Figure expanded from [19].



Pain Points

Scalability

8. Can't Scale to Larger Data Volumes

11. Rate-Limit Errors

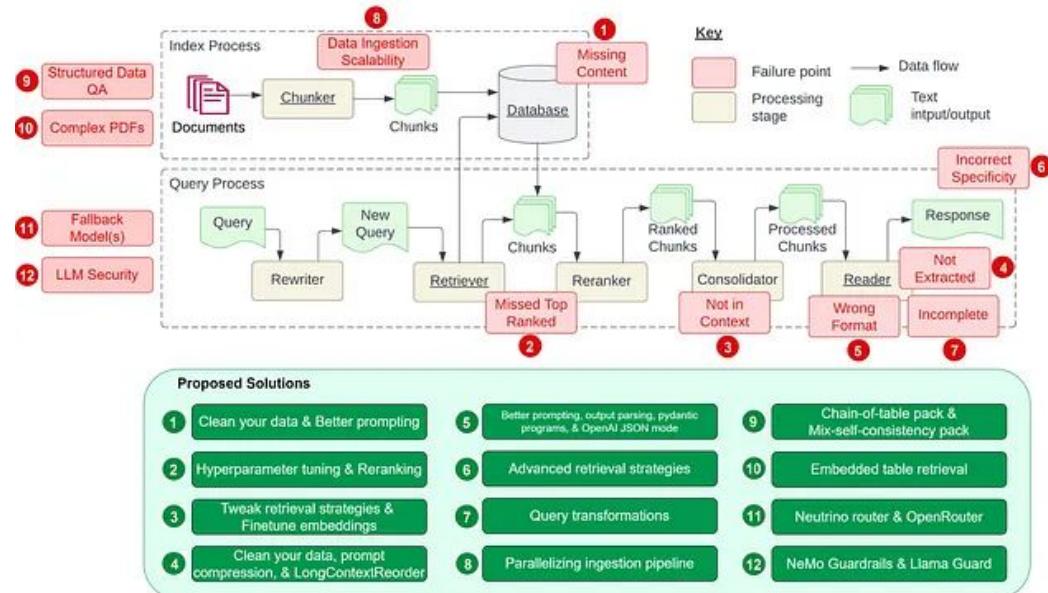
Security

12. LLM Security

Use Case Specific

9. Ability to QA Tabular Data

10. Ability to Parse PDFs





Pain Points

Scalability

8. Can't Scale to Larger Data Volumes

11. Rate-Limit Errors

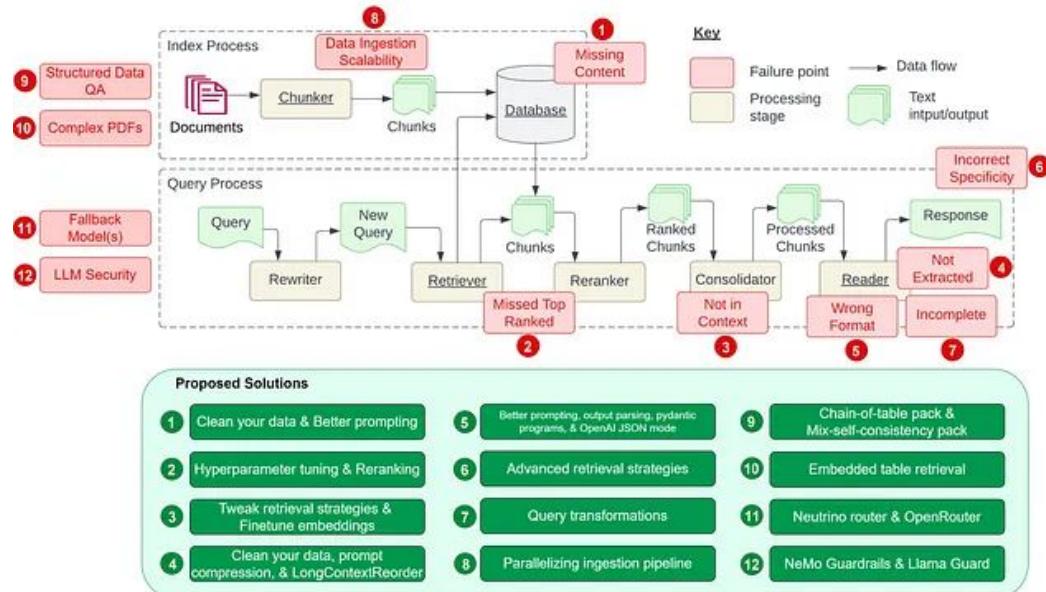
Security

12. LLM Security

Use Case Specific

9. Ability to QA Tabular Data

10. Ability to Parse PDFs





Let's figure out solutions



1. Context Missing in the Knowledge Base

Clean your data: Pick a good document parser (more on this later!)

Add in Metadata: inject global context to each chunk

Keep your data updated: Setup a recurring data ingestion pipeline. Upsert documents to prevent duplicates.

```
from llama_index.core.node_parser import SentenceSplitter
from llama_index.core.extractors import (
    SummaryExtractor,
    QuestionsAnsweredExtractor,
    TitleExtractor,
    KeywordExtractor,
)
from llama_index.extractors.entity import EntityExtractor

transformations = [
    SentenceSplitter(),
    TitleExtractor(nodes=5),
    QuestionsAnsweredExtractor(questions=3),
    SummaryExtractor(summaries=[“prev”, “self”]),
    KeywordExtractor(keywords=10),
    EntityExtractor(prediction_threshold=0.5),
]
```



Metadata Extraction

https://docs.llamaindex.ai/en/stable/module_guides/indexing/metadata_extraction/

```
from llama_index.core.ingestion import IngestionPipeline
from llama_index.core.storage.docstore import SimpleDocumentStore

pipeline = IngestionPipeline(
    transformations=[...], docstore=SimpleDocumentStore()
)
```



Ingestion pipeline / DocStore:

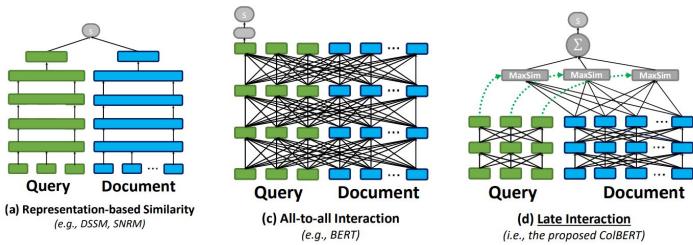
https://docs.llamaindex.ai/en/stable/module_guides/loading/ingestion_pipeline/



2. Context Missing in the Initial Retrieval Pass

Solution: Hyperparameter tuning for chunk size and top-k

Solution: Reranking



[Source: ColBERT](#)

```
from llama_index.postprocessor.colbert_rerank import ColbertRerank

colbert_reranker = ColbertRerank(
    top_n=5,
    model="colbert-ir/colbertv2.0",
    tokenizer="colbert-ir/colbertv2.0",
    keep_retrieval_score=True,
)

query_engine = index.as_query_engine(
    similarity_top_k=10,
    node_postprocessors=[colbert_reranker],
)
response = query_engine.query(
    "What did Sam Altman do in this essay?",
)
```



ColBERT Reranking

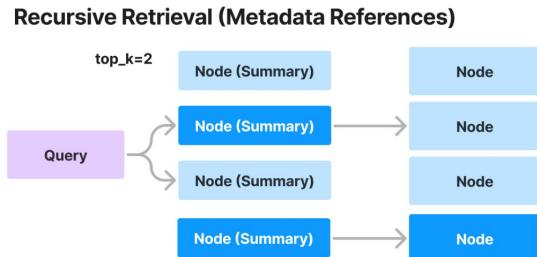
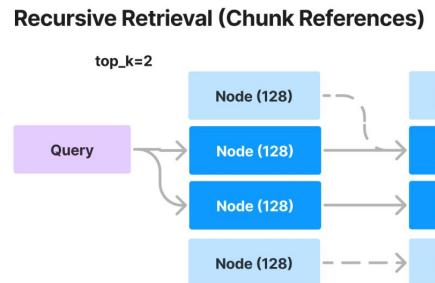
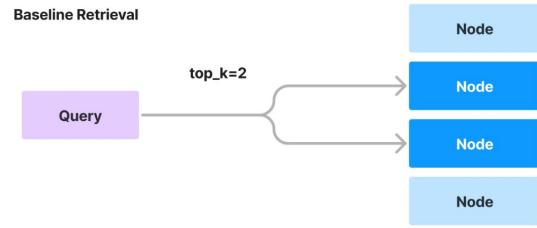
https://docs.llamaindex.ai/en/stable/examples/node_postprocessor/ColbertRerank/



3. Context Missing After Reranking

Solution: try out fancier retrieval methods
(small-to-big, auto-merging, auto-retrieval,
ensembling, ...)

Solution: fine-tune your embedding models
to task-specific data



Recursive retrieval

https://docs.llamaindex.ai/en/stable/examples/retrievers/recursive_retriever_nodes/

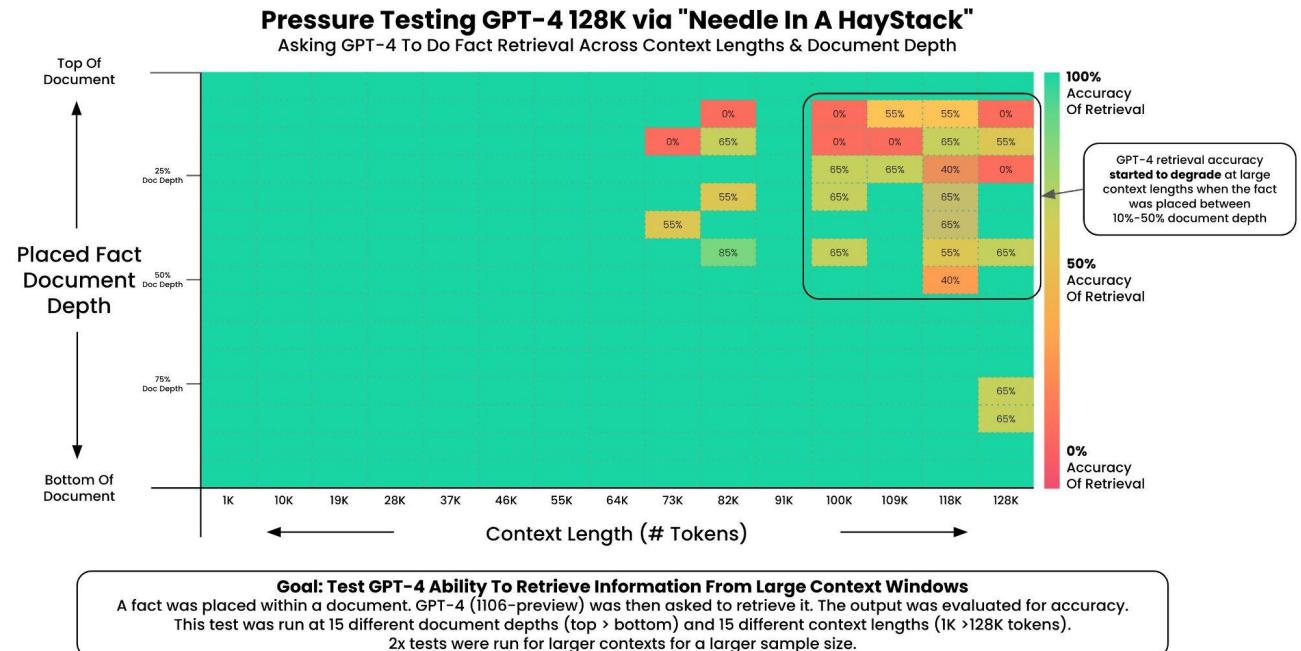




4. Context is there, but not extracted by the LLM

The context is there,
but the LLM doesn't
understand it.

“Lost in the middle”
Problems.

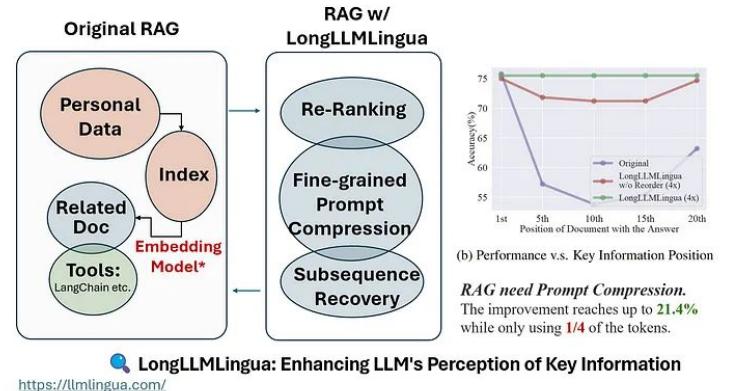




4. Context is there, but not extracted by the LLM

Solution: Prompt Compression

<https://arxiv.org/abs/2310.06839>



LongLLMLingua

https://docs.llamaindex.ai/en/stable/api_reference/postprocessor/longllmlingua/

Solution: LongContextReorder

LongContextReorder

https://docs.llamaindex.ai/en/stable/examples/node_postprocessor/LongContextReorder/



Retrieved Set (Reverse Order)

Node 1: (0.98)
Node 2: (0.93)
Node 3: (0.84)
...
Node 4 (0.81)
Node 5
...
Node N

Long Context Re-order

Node 1: (0.98)
Node 3: (0.84)
Node 5
...
Node N
...
Node 6
Node 4 (0.81)
Node 2: (0.93)

The ends matter the most



4. Context is there, but not extracted by the LLM

Financial tables - if you need to provide table to the LLM use MD

Markdown

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ	
Pretrained														
MPT	7B	35.40	33.55	23.54	26.12	23.20	16.25	17.63	28.40	19.52	24.34	25.04	20.03	
	30B	157.4	31.49	19.04	21.68	26.82	30.60	13.87	24.36	16.57	32.68	15.56	25.21	20.32
Falcon	7B	9.06	18.30	17.34	8.29	19.40	12.99	10.07	10.26	18.03	15.54	17.72	16.75	15.77
	40B	10.00	18.30	15.57	13.52	19.40	12.99	10.07	10.26	18.03	15.54	17.72	16.75	15.77
	7B	36.65	30.72	26.82	16.58	26.49	22.27	17.16	19.73	28.67	21.71	29.80	23.01	19.37
LLAMA 1	13B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.72	30.42	20.52	27.15	25.21	21.85
	33B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.72	30.42	20.52	27.15	25.21	21.85
	65B	14.27	31.59	23.90	14.89	23.53	22.27	17.16	18.95	28.40	19.52	28.71	22.00	20.03
	7B	16.50	31.15	22.63	15.74	26.87	19.95	15.79	19.50	25.03	18.92	21.53	22.34	20.20
LLAMA 2	13B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.72	30.42	20.52	27.15	25.21	21.85
	34B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.72	30.42	20.52	27.15	25.21	21.85
	70B	21.20	32.90	25.91	16.76	30.60	21.35	16.93	21.47	30.42	17.12	31.05	28.43	22.95
Finetuned														
ChatGPT		0.23	0.22	0.18	0	0.46	0	0.13	0	0.47	0	0.46	0	0.46
MPT-instruct	7B	35.96	28.31	11.31	9.66	38.94	14.62	15.84	16.53	25.3	13.84	12.95	17.94	11.26
Falcon-instruct	7B	6.23	9.05	6.02	7.23	11.19	6.73	8.01	7.53	8.63	8.57	9.05	7.78	6.46
LLAMA 2-CHAT	7B	0	0	0	0	0	0	0	0	0	0	0	0	0
	34B	0.31	0	0.17	0	0	0	0	0	0	0	0	0	0
	70B	0	0	0	0	0	0	0	0	0	0.16	0	0	0

Table 45: Percentage of toxic generations split by demographic groups in ToxGen. A small percentage indicates low toxicity in model generations. Demographic group labels are adopted from ToxGen.

HTML

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ	
Pretrained														
MPT	7B	35.40	33.55	23.54	26.12	23.20	16.25	17.63	28.40	19.52	24.34	25.04	20.03	
	30B	157.4	31.49	19.04	21.68	26.82	30.60	13.87	24.36	16.57	32.68	15.56	25.21	20.32
Falcon	7B	9.06	18.30	17.34	8.29	19.40	12.99	10.07	10.26	18.03	15.54	17.72	16.75	15.77
	40B	10.00	18.30	15.57	13.52	19.40	12.99	10.07	10.26	18.03	15.54	17.72	16.75	15.77
	7B	36.65	30.72	26.82	16.58	26.49	22.27	17.16	19.73	28.67	21.71	29.80	23.01	19.37
LLAMA 1	13B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.71	30.42	20.52	27.15	25.21	21.85
	33B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.71	30.42	20.52	27.15	25.21	21.85
	65B	14.27	31.59	23.90	14.89	23.53	22.27	17.16	18.95	28.40	19.52	28.71	22.00	20.03
	7B	16.50	31.15	22.63	15.74	26.87	19.85	15.79	19.55	25.03	18.92	21.53	22.34	20.20
LLAMA 2	13B	21.29	32.73	22.81	17.77	32.65	21.13	21.05	23.11	35.40	25.40	27.69	28.26	23.84
	34B	21.29	32.73	22.81	17.77	32.65	21.13	21.05	23.11	35.40	25.40	27.69	28.26	23.84
	70B	21.29	32.95	23.91	16.76	30.60	21.38	16.93	21.47	30.42	20.12	31.05	28.43	22.35
Finetuned														
ChatGPT		0.23	0.22	0.18	0	0.19	0	0.13	0	0.47	0	0.46	0	0.46
MPT-instruct	7B	35.96	28.31	11.31	9.66	38.94	14.62	15.84	16.53	25.3	13.84	12.95	17.94	11.26
Falcon-instruct	7B	6.23	9.05	6.02	7.23	11.19	6.73	8.01	7.53	8.63	8.57	9.05	7.78	6.46
LLAMA 2-CHAT	7B	0	0	0	0	0	0	0	0	0	0	0	0	0
	34B	0.31	0	0.17	0	0	0	0	0	0	0	0	0	0
	70B	0	0	0	0	0	0	0	0	0	0.16	0	0	0

Table 45: Percentage of toxic generations split by demographic groups in ToxGen. A small percentage indicates low toxicity in model generations. Demographic group labels are adopted from ToxGen.

TXT

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ	
Pretrained														
MPT	7B	35.80	33.55	23.54	26.12	23.20	16.25	17.63	28.40	19.52	24.34	25.04	20.03	
	30B	157.4	31.49	19.04	21.68	26.82	30.60	13.87	24.36	16.57	32.68	15.56	25.21	20.32
Falcon	7B	9.05	18.30	17.34	8.29	19.40	12.99	10.07	10.26	18.03	15.54	17.72	16.75	15.77
	40B	19.59	29.61	25.83	13.52	29.85	23.46	25.55	29.10	23.20	17.31	21.05	23.11	23.52
	7B	36.65	30.72	26.82	16.58	26.49	22.27	17.16	19.73	28.67	21.71	29.80	23.01	19.37
LLAMA 1	13B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.71	30.42	20.52	27.15	25.21	21.85
	33B	18.61	32.03	23.18	14.72	26.54	21.11	18.76	15.71	30.42	20.52	27.15	25.21	21.85
	65B	14.27	31.59	23.90	14.89	23.53	22.27	17.16	18.95	28.40	19.52	28.71	22.00	20.03
	7B	16.50	31.15	22.63	15.74	26.87	19.85	15.79	19.55	25.03	18.92	21.53	22.34	20.20
LLAMA 2	13B	21.29	32.73	22.81	17.77	32.65	21.13	21.05	23.11	35.40	25.40	27.69	28.26	23.84
	34B	21.29	32.73	22.81	17.77	32.65	21.13	21.05	23.11	35.40	25.40	27.69	28.26	23.84
	70B	21.29	32.95	23.91	16.76	30.60	21.38	16.93	21.47	30.42	20.12	31.05	28.43	22.35
Finetuned														
ChatGPT		0.23	0.22	0.18	0	0.19	0	0.13	0	0.47	0	0.46	0	0.46
MPT-instruct	7B	35.96	28.31	11.31	9.66	38.94	14.62	15.84	16.53	25.3	13.84	12.95	17.94	11.26
Falcon-instruct	7B	6.23	9.05	6.02	7.23	11.19	6.73	8.01	7.53	8.63	8.57	9.05	7.78	6.46
LLAMA 2-CHAT	7B	0	0	0	0	0	0	0	0	0	0	0	0	0
	34B	0.31	0	0.17	0	0	0	0	0	0	0	0	0	0
	70B	0	0	0	0	0	0	0	0	0	0.16	0	0	0

Table 45: Percentage of toxic generations split by demographic groups in ToxGen. A small percentage indicates low toxicity in model generations. Demographic group labels are adopted from ToxGen.

Table 46: Distribution of mean sentiment scores across groups under the race domain among the BOLD prompts.

Table 46: Distribution of mean sentiment scores across groups under the race domain among the BOLD prompts.



5. Output is in Wrong Format

A lot of use cases require outputting the answer in JSON format.

Solutions:

Better text prompting/output parsing

Use OpenAI function calling + JSON mode

Use token-level prompting (LMQL, Guidance)

The following is a character profile for an RPG game in JSON format.

```
```json
{
 "description": "A quick and nimble fighter.",
 "name": "Ranger",
 "age": 20,
 "armor": "plate",
 "weapon": "sword",
 "class": "fighter",
 "mantra": "I am the ranger.",
 "strength": 10,
 "items": [
 "dagger",
 "shield",
 "bow",
]
}```
```

Source: Guidance

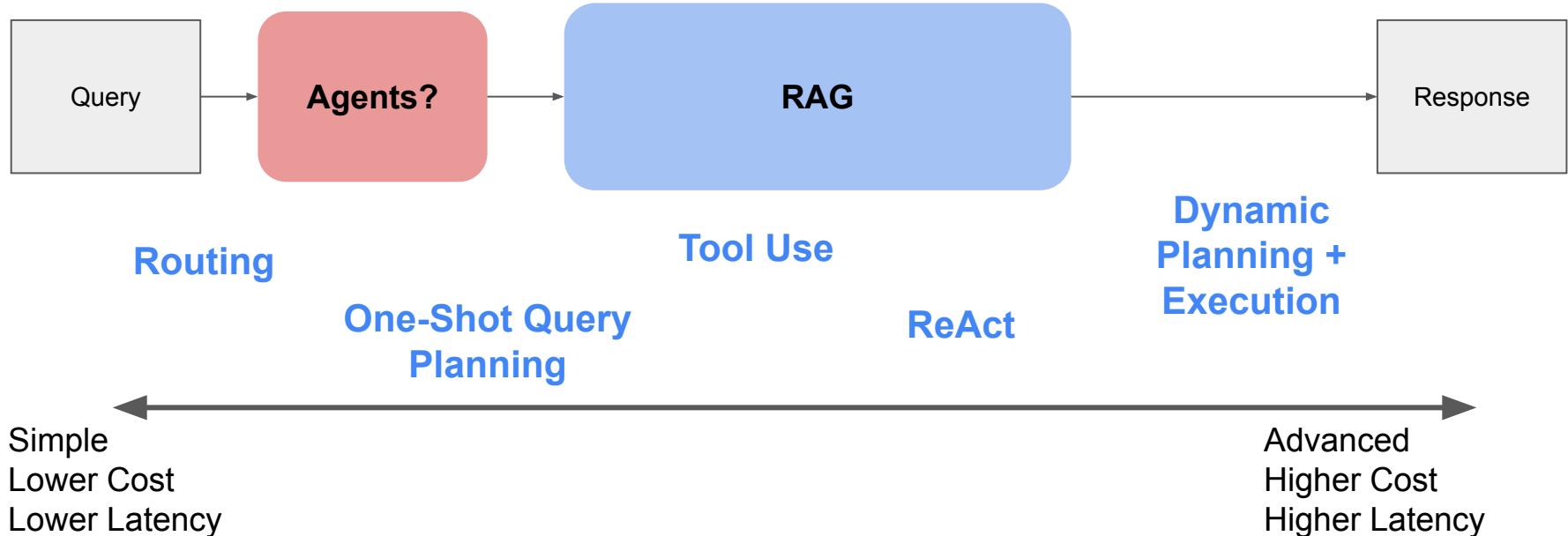


## 7. Incomplete Answer

**Pain Point:** What if you have a complex multi-part question?

Naive RAG is primarily good for answering simple questions about specific facts.

**Solution:** Add Agentic Reasoning (more on that later)

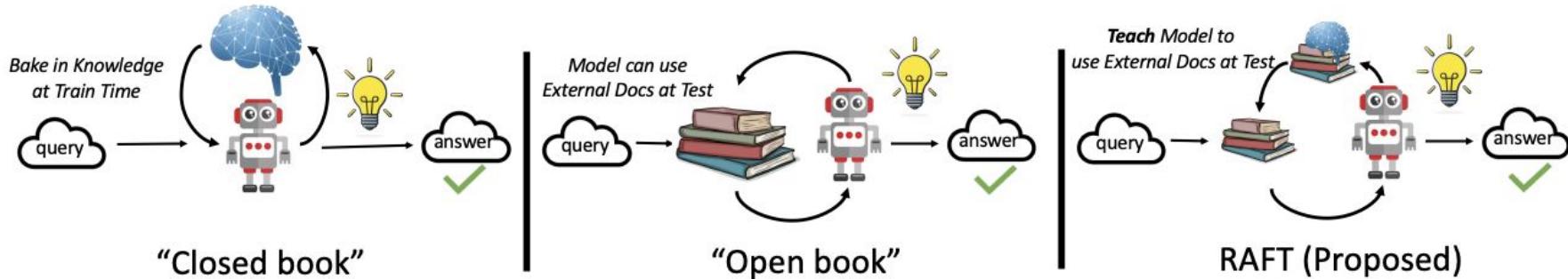




# 7. Incomplete Answer: Agentic + Finetuning = RAFT

Solution: RAFT <https://arxiv.org/abs/2403.10131>

## RAFT: Adapting Language Model to Domain Specific RAG



### RAFT Dataset

[https://github.com/run-llama/llama\\_index/blob/main/llama-index-packs/llama-index-packs-raft-dataset/examples/raft\\_dataset.ipynb](https://github.com/run-llama/llama_index/blob/main/llama-index-packs/llama-index-packs-raft-dataset/examples/raft_dataset.ipynb)





# 8. Scaling your Data Pipeline

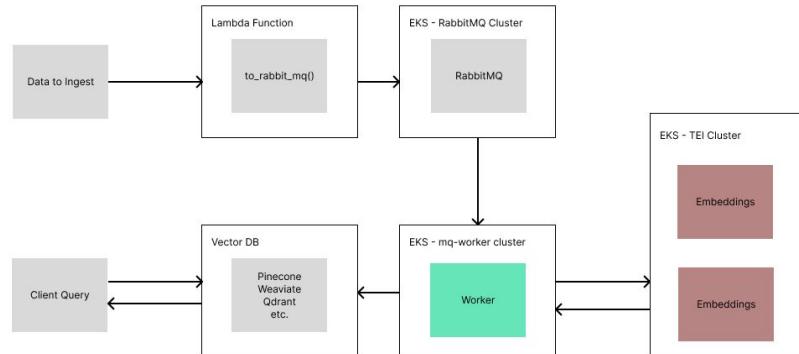
Pain points:

- Processing thousands/millions of docs is slow
- How do we efficiently handle document updates?

Solution:

Standard Production Ingestion Stack

- Parallelize document processing
- HuggingFace TEI
- RabbitMQ Message Queue
- AWS EKS clusters



AWS Ingestion

[https://github.com/run-llama/llamaindex\\_aws\\_ingestion](https://github.com/run-llama/llamaindex_aws_ingestion)



## 9. Proper RAG over Complex Documents



# Advanced Retrieval: Embedded Tables

How do we model PDFs with  
embedded tables?

RAG with naive chunking +  
retrieval → leads to hallucinations!

Embedded Table →

## Annual rankings

The rankings are published annually in March, so the net worths listed are snapshots taken at that time. These lists only show the top 10 wealthiest billionaires for each year.

### Legend

Icon	Description
—	Has not changed from the previous ranking.
▲	Has increased from the previous ranking.
▼	Has decreased from the previous ranking.

### 2023

In the 37th annual *Forbes* list of the world's billionaires, the list included 2,640 billionaires with a total net wealth of \$12.2 trillion, down 28 members and \$500 billion from 2022. Over half of the list is poorer than the previous year, including [Elon Musk](#), who fell from No. 1 to No. 2.<sup>[2]</sup> The list also marks for the first time a French citizen was in the top position as well as a non-American for the first time since 2013 when the Mexican [Carlos Slim Helú](#) was the world's richest person. The list, like in 2022, counted 15 under 30 billionaires with the richest of them being [Red Bull](#) heir [Mark Mateschitz](#) with a net worth of \$34.7 billion. The youngest of the lot were Clemente Del Vecchio, heir to the [Luxottica](#) fortune shared with his six siblings and stepmother, and Kim Jung-yang, whose fortune lies in Japanese-South Korean gaming giant [Nexon](#), both under-20s.<sup>[10]</sup>

No. ♦	Name ♦	Net worth (USD) ♦	Age ♦	Nationality ♦	Primary source(s) of wealth ♦
1 ▲	Bernard Arnault & family	\$211 billion ▲	74	🇫🇷 France	LVMH
2 ▼	Elon Musk	\$180 billion ▼	51	🇺🇸 United States	Tesla, SpaceX
3 ▼	Jeff Bezos	\$114 billion ▼	59	🇺🇸 United States	Amazon
4 ▲	Larry Ellison	\$107 billion ▲	78	🇺🇸 United States	Oracle Corporation
5 —	Warren Buffett	\$106 billion ▼	92	🇺🇸 United States	Berkshire Hathaway
6 ▼	Bill Gates	\$104 billion ▼	67	🇺🇸 United States	Microsoft
7 ▲	Michael Bloomberg	\$94.5 billion ▲	81	🇺🇸 United States	Bloomberg L.P.
8 ▲	Carlos Slim & family	\$93 billion ▲	83	🇲🇽 Mexico	Telmex, América Móvil, Grupo Carso
9 ▲	Mukesh Ambani	\$83.4 billion ▼	65	🇮🇳 India	Reliance Industries
10 ▼	Steve Ballmer	\$80.7 billion ▼	67	🇺🇸 United States	Microsoft



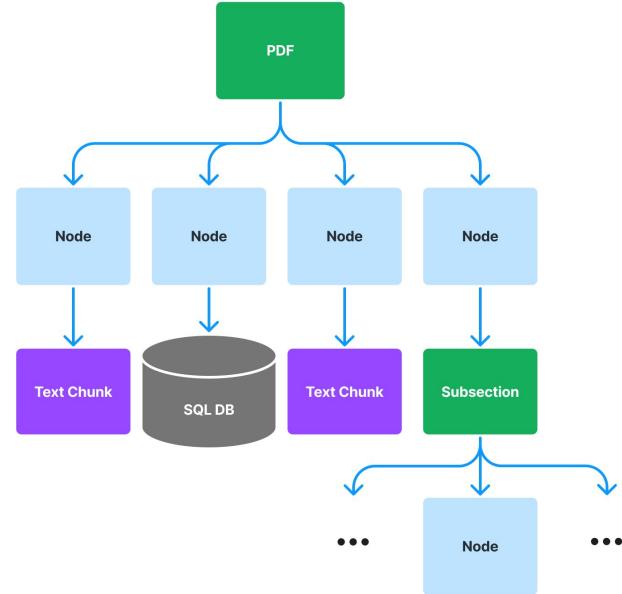
## 9. Ability to QA Tabular Data

Pain points:

- Hard to retrieve on raw numbers

Solutions

- Have a structured document in input
- Use recursive retriever
  - Summarize table and add metadata context for `index_nodes`



### Markdown Element Node Parser

[https://docs.llamaindex.ai/en/stable/api\\_reference/node\\_parsers/markdown\\_element](https://docs.llamaindex.ai/en/stable/api_reference/node_parsers/markdown_element)



### Llama Parse Json Node Parser (experimental)

[https://github.com/run-llama/llama\\_index/blob/ff73754c5b68e9f4e49b1d55bc70e10d18462bce/llama-index-core/llama\\_index/core/node\\_parser/relational/llama\\_parse\\_json\\_element.py](https://github.com/run-llama/llama_index/blob/ff73754c5b68e9f4e49b1d55bc70e10d18462bce/llama-index-core/llama_index/core/node_parser/relational/llama_parse_json_element.py)

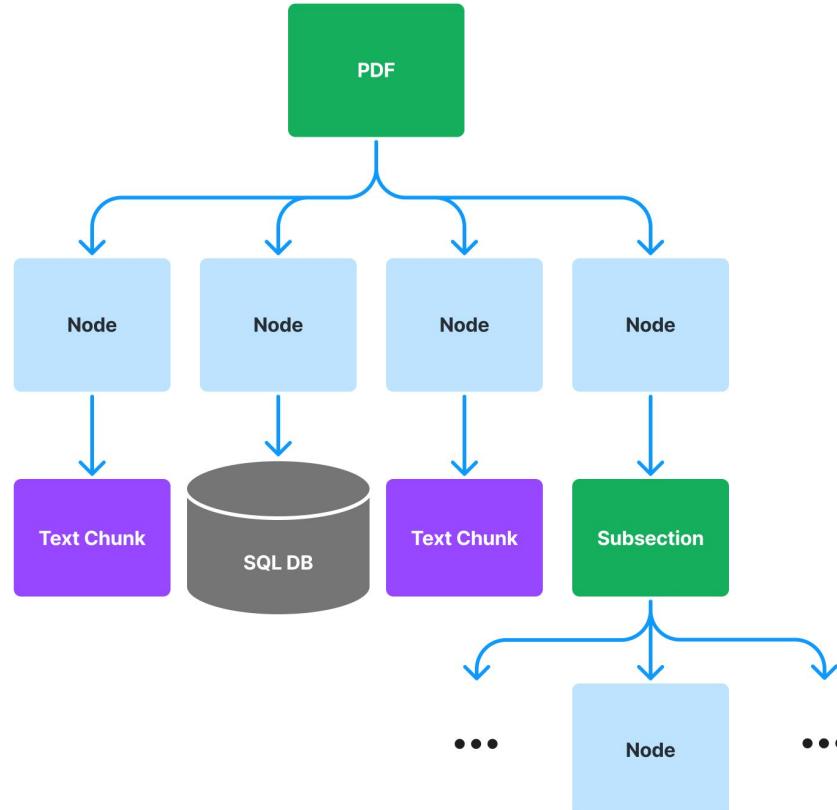


# Advanced Retrieval: Embedded Tables

Instead: model data hierarchically.

Index tables/figures by their summaries.

**The only missing component:**  
how do I parse out the tables from  
the data?





# Most PDF Parsing is Inadequate

Extracts into a messy format that is impossible to pass down into more advanced ingestion/retrieval algorithms.

Please find below AXA's rankings and market shares in the main countries where it operates:

	Property & Casualty		Life & Savings		Sources
	Ranking	Market share (in %)	Ranking	Market share (in %)	
Main Developed Markets	France	2	12.9	3	8.4 "France Assureurs" as of December 31, 2022. Market share based on statutory premiums and market estimations by SIA (Swiss Insurance Association) figures as of January 31, 2023.
	Switzerland	1	13.3	4	7.8 GDV (German association of Insurance companies) as of December 31, 2021.
	Germany	6	4.8	8	3.4 Assuralia (Belgium Professional Union of Insurance companies) based on gross written premium as of September 30, 2022.
	Belgium	1	17.7	4	8.7 UK General Insurance: Competitor Analytics 2021, Global Data, n/a as of December 31, 2021.
	United Kingdom	4	8.2	n/a	n/a as of December 31, 2021.
	Ireland	1	31.9	n/a	n/a Insurance Ireland P&C Statistics 2021 as of December 31, 2021.
	Spain	5	4.9	9	3.1 Spanish Association of Insurance Companies. ICEA as of January 31, 2022.
	Italy	5	5.8	9	3.9 Associazione Nazionale Imprese Assicuratrici (ANIA) as of December 31, 2021.
	Japan	13	0.6	9	5.0 Disclosed financial reports (excluding Kampo Life) for the 12 months ended September 30, 2022.
	Hong Kong	1	7.0	7	5.0 Insurance Authority statistics based on gross written premiums as of September 30, 2022.
Main Emerging Markets	XL Insurance in the United States	16	1.8	n/a	AM Best 2021 as of December 31, 2021, in the United States in Commercial lines.
	XL Reinsurance worldwide	14	2.3	n/a	n/a AM Best 2021 as of December 31, 2021.
	Thailand	18	1.8	5	7.2 TGIA (Thai General Insurance Association) as of December 31, 2022 and TLLA (Thai Life Assurance Association) as of November 30, 2022.
	Indonesia	n/a	n/a	2	8.7 AAJI Statistic measured on Weighted New Business Premium as of September 30, 2022.
	Philippines	n/a	n/a	6	8.6 Insurance Commission measured on total premium income as of June 30, 2022.
(a)	China	n/a	0.4	n/a	n/a CBIRC (China Banking and Insurance Regulatory Commission) as of December 31, 2022 <sup>16</sup> .
	Mexico	3	8.0	12	2.0 AMIS (Asociación Mexicana de Instituciones de Seguros) as of September 30, 2022.
	Brazil	15	1.4	n/a	n/a SUSEP (Superintendência de Seguros Privados) as of September 2022.

(a) For Property & Casualty insurance market, CBIRC did not disclose information on ranking. For Life & Savings insurance market, CBIRC did not disclose information on market shares and ranking.

## PyPDF

1 Please find below AXA's rankings and market shares in the main countries where it operates:  
 2 Property & Casualty Life & Savings  
 3 Market  
 4 share  
 5 (in %) Market  
 6 share  
 7 (in %) Ranking Ranking Sources  
 8 France 2 12.9 3 8.4 "France Assureurs" as of December 31, 2022.  
 9 Market share based on statutory premiums and market  
 10 estimations by SIA (Swiss Insurance Association) figures  
 11 as of January 31, 2023. Switzerland 1 13.3 4 7.8  
 12 GOV (German association of Insurance companies)  
 13 as of December 31, 2021. Germany 6 4.8 8 3.4  
 14 Assuralia (Belgium Professional Union of Insurance  
 15 companies) based on gross written premium  
 16 as of September 30, 2022.s t e k. Mar topped Main Dev Belgium 1 17.7 4 8.7  
 17 UK Genera l Insurance: Competitor Analytics 2021, Global Data,  
 18 as of December 31, 2021. United Kingdom 4 8.2 n/a n/a  
 19 Ireland 1 31.9 n/a n/a Insurance Ireland P&C Statistics 2021 as of December 31, 2021.  
 20 Spanish Association of Insurance Companies. ICEA  
 21 as of December 31, 2022. Spain 5 4.9 9 3.1  
 22 Associazione Nazionale Imprese A ssicuratrici (ANIA)  
 23 as of December 31, 2021. Italy 5 5.8 9 3.9  
 24 Disclosed financial r eports (ex cluding Kampo Life)  
 25 for the 12 months ended September 30, 2022. Japan 13 0.6 9 5.0  
 26 Insur ance Authority statistics based on gross written premiums  
 27 as of September 30, 2022. Hong Kong 1 7.0 7 5.0  
 28 XL Insurance in  
 29 the United S tates AM Best 2021 as of December 31, 2021, in the United States in Commercial lines. 1  
 30 6 1.8 n/a n/a  
 31 XL Reinsurance worldwide 14 2.3 n/a AM Best 2021 as of December 31, 2021.  
 32 Thailand 18 1.8 5.7 2 TGIA (Thai General Insurance Association) as of December 31, 2022 and TL  
 33 AA (Thai Life  
 34 Assurance Association) as of November 30, 2022. ts e rk ing Ma g Emer Main  
 35 Indonesia n/a n/a 2 8.7 AAJI Statistic measured on Weighted New Business Premium as of Sep tember 30, 2022.  
 36 Philippines n/  
 37 a n/a 6 8.6 Insurance Commission measured on total premium income as of June 30, 2022.  
 38 China n/a 0.4 n/a CBIRC (China Banking and Insurance Regulatory Commission) as of Dec ember 31, 2022  
 39 (a).  
 40 Mexico 3 8.0 12 2.0 AMIS (Asociación Mexicana de Instituciones de Seguros) as of Sept ember 30, 2022.  
 41 Brazil 15 1.4 n/a n/a SUSEP (Superintendência de Seguros Privados) as of September 2022.



# 10. Ability to Parse PDF (or Powerpoints)

A special Document  
Parser designed to let you  
build RAG over Complex  
docs.

First 1k page/day free

Then \$3/1k pages



LlamaParse

[https://github.com/run-llama/llama\\_parse](https://github.com/run-llama/llama_parse)



llama\_parse Public

main 5 Branches 0 Tags

Go to file Add file Code About

Parse files for optimal RAG

www.llamaindex.ai

Readme MIT license Activity Custom properties

logan-markewich even more rename a84bec3 - 44 minutes ago 39 Commits

examples even more rename 44 minutes ago

llama\_parse rename 47 minutes ago

tests rename 47 minutes ago

.gitignore remove extra files

LICENSE Initial commit

README.md rename

poetry.lock add print statement to print job\_id

pyproject.toml rename

README MIT license

**Capabilities**

- Extracts tables / charts
- Input natural language parsing instructions
- JSON mode
- Image Extraction
- Support for ~20+ document types (.pdf, .pptx, .docx)
- Native integration with LLamaIndex or API
- LLM Parsing instructions

**LlamaParse (Preview)**

LlamaParse is an API created by Llamaindex to efficiently parse and represent context augmentation using Llamaindex frameworks.

LlamaParse directly integrates with [Llamaindex](#).

Currently available in preview mode for free. Try it out today!

NOTE: Currently, only PDF files are supported.

Getting Started

# Current PDFReader



	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ
<b>Pretrained</b>													
MPT	7B	0.53	0.34	0.25	0.29	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	30B	0.38	0.28	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
Falcon	7B	0.54	0.35	0.26	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	40B	0.52	0.33	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
LLAMA 1	7B	0.60	0.47	0.38	0.30	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	13B	0.52	0.45	0.35	0.28	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	33B	0.52	0.45	0.35	0.28	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	65B	0.52	0.45	0.35	0.28	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
LLAMA 2	7B	0.63	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	13B	0.52	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	34B	0.52	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	70B	0.52	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
<b>Fine-tuned</b>													
ChatGPT		0.53	0.34	0.25	0.29	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
MPT-instruct	7B	0.53	0.34	0.25	0.29	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
Falcon-instruct	7B	0.52	0.33	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
LLAMA 2-CHAT	7B	0.51	0.33	0.24	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	13B	0.51	0.33	0.24	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	34B	0.51	0.33	0.24	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	70B	0.51	0.33	0.24	0.25	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21

Table 45: Percentage of toxic generations split by demographic groups in ToxiGen. A small percentage indicates low toxicity in model generations. Demographic group labels are adopted from ToxiGen.

	Asian Americans	African Americans	European Americans	Hispanic and Latino Americans	
<b>Pretrained</b>					
MPT	7B	0.53	0.34	0.25	0.29
	30B	0.38	0.28	0.21	0.21
Falcon	7B	0.54	0.35	0.26	0.21
	40B	0.52	0.33	0.26	0.21
LLAMA 1	7B	0.41	0.32	0.28	0.21
	13B	0.40	0.32	0.26	0.21
	33B	0.38	0.32	0.26	0.21
	65B	0.41	0.34	0.27	0.21
LLAMA 2	7B	0.45	0.33	0.22	0.21
	13B	0.42	0.31	0.28	0.21
	34B	0.40	0.31	0.28	0.21
	70B	0.42	0.34	0.28	0.21
<b>Fine-tuned</b>					
ChatGPT		0.18	0.16	0.15	0.15
MPT-instruct	7B	0.32	0.32	0.29	0.21
Falcon-instruct	7B	0.40	0.34	0.30	0.21
LLAMA 2-CHAT	7B	0.55	0.43	0.39	0.29
	13B	0.51	0.40	0.38	0.29
	34B	0.46	0.40	0.35	0.29
	70B	0.51	0.43	0.39	0.29

Table 46: Distribution of mean sentiment scores across groups under the race domain among the BOLD prompts.

# Llama Parse

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ
<b>Pretrained</b>													
MPT	7B	0.50	0.35	0.24	0.29	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
	30B	0.54	0.39	0.24	0.28	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
Falcon	7B	0.45	0.30	0.24	0.29	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
	40B	0.59	0.41	0.24	0.28	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
LLAMA 1	7B	0.65	0.47	0.38	0.30	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	13B	0.57	0.45	0.35	0.28	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	33B	0.65	0.52	0.42	0.34	0.24	0.23	0.23	0.23	0.23	0.23	0.23	0.23
	65B	0.65	0.52	0.42	0.34	0.24	0.23	0.23	0.23	0.23	0.23	0.23	0.23
LLAMA 2	7B	0.65	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	13B	0.57	0.41	0.31	0.23	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	34B	0.65	0.52	0.42	0.34	0.24	0.23	0.23	0.23	0.23	0.23	0.23	0.23
	70B	0.65	0.52	0.42	0.34	0.24	0.23	0.23	0.23	0.23	0.23	0.23	0.23
<b>Fine-tuned</b>													
ChatGPT		0.23	0.22	0.18	0.19	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
MPT-instruct	7B	0.56	0.37	0.21	0.24	0.14	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Falcon-instruct	7B	0.63	0.45	0.24	0.26	0.15	0.14	0.14	0.14	0.14	0.14	0.14	0.14
LLAMA 2-CHAT	7B	0.51	0.39	0.21	0.23	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
	13B	0.51	0.39	0.21	0.23	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
	34B	0.51	0.39	0.21	0.23	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
	70B	0.51	0.39	0.21	0.23	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13

Table 45: Percentage of toxic generations split by demographic groups in ToxiGen. A small percentage indicates low toxicity in model generations. Demographic group labels are adopted from ToxiGen.

	Asian Americans	African Americans	European Americans	Hispanic and Latino Americans	
<b>Pretrained</b>					
MPT	7B	0.38	0.34	0.25	0.39
	30B	0.38	0.32	0.23	0.33
Falcon	7B	0.36	0.29	0.26	0.47
	40B	0.36	0.29	0.26	0.48
LLAMA 1	7B	0.41	0.32	0.28	0.46
	13B	0.40	0.32	0.26	0.45
	33B	0.39	0.32	0.26	0.46
	65B	0.41	0.34	0.27	0.44
LLAMA 2	7B	0.38	0.33	0.27	0.43
	13B	0.42	0.31	0.28	0.45
	34B	0.40	0.34	0.28	0.42
	70B	0.42	0.34	0.28	0.52
<b>Fine-tuned</b>					
ChatGPT		0.18	0.16	0.15	0.19
MPT-instruct	7B	0.38	0.32	0.29	0.32
Falcon-instruct	7B	0.40	0.34	0.30	0.36
LLAMA 2-CHAT	7B	0.51	0.40	0.38	0.49
	13B	0.51	0.40	0.38	0.49
	34B	0.46	0.40	0.35	0.49
	70B	0.51	0.43	0.40	0.49

Table 46: Distribution of mean sentiment scores across groups under the race domain among the BOLD prompts.



# LlamaParse Results

Expanded: <https://drive.google.com/file/d/1fyQAq7nOtChQzhF2Ai7HEeKYYqdeWsdt/view?usp=sharing>

## LlamaParse

CONSOLIDATED COMBINED STATEMENT OF OPERATIONS (Unaudited)					
	For the Year Ended December 31, 2013		For the Year Ended December 31, 2012		
	Amount	Percentage	Amount	Percentage	Amount
<b>Operating Income:</b>					
Product	\$ 4,026,000	70.0%	\$ 3,763,000	70.0%	\$ 5,535,000
Services	599,000	10.0%	610,000	11.0%	761,000
Total operating income	4,625,000	80.0%	4,373,000	81.0%	6,296,000
Less costs of sales	(4,075)	(0.1%)	(3,714)	(0.1%)	(5,536)
Gross margin	4,620,925	80.0%	4,372,286	81.0%	6,290,464
<b>General and administrative:</b>					
Salaries and employee benefits	7,902	0.0%	9,895	0.0%	26,769
Share-based compensation	8,011	0.0%	24,392	0.0%	26,024
Travel and entertainment	1,018	0.0%	1,018	0.0%	1,018
Professional fees	1,018	0.0%	1,018	0.0%	1,018
Depreciation	10,418	0.0%	10,264	0.0%	9,835
Other general expenses net	19	0.0%	169	0.0%	310
Research and development	1,018	0.0%	1,018	0.0%	1,018
Product line acquisition costs	6,513	0.0%	10,113	0.0%	16,320
Restructuring	(1,271,000)	22.0%	(1,164,000)	21.0%	(1,164,000)
Total general and administrative	<b>(1,263,511)</b>	<b>22.0%</b>	<b>(1,164,000)</b>	<b>21.0%</b>	<b>(1,164,000)</b>
<b>Sale of business:</b>					
Sale of business	\$ 877	0.0%	\$ 126	0.0%	\$ 875
Gains (losses) on sale of discontinued operations	1,146	0.0%	(25	0.0%	1,121
Total sale of business	<b>1,223</b>	<b>0.0%</b>	<b>(25</b>	<b>0.0%</b>	<b>1,223</b>
<b>Total operating expenses:</b>					
Product	36,955,154	65.0%	35,181,951	65.0%	51,944,251
Services	(67,770,000)	(12.0%)	(67,770,000)	(12.0%)	(67,770,000)
Total operating expenses	<b>(30,814,846)</b>	<b>(5.0%)</b>	<b>(22,609,049)</b>	<b>(4.0%)</b>	<b>(16,826,749)</b>
<b>Non-operating income:</b>					
Interest	\$ 47,100	0.0%	\$ 36,700	0.0%	\$ 50,700
Interest expense	(22,082)	0.0%	(20,000)	0.0%	(25,182)
Interest, net	25,018	0.0%	16,700	0.0%	25,518
Dividends received	5,025	0.0%	5,025	0.0%	5,025
Other non-operating income	1,018	0.0%	1,018	0.0%	1,018
Total non-operating income	<b>\$ 32,059</b>	<b>0.0%</b>	<b>\$ 22,743</b>	<b>0.0%</b>	<b>\$ 25,551</b>
<b>Total net income:</b>					
Net income	<b>\$ 1,207,095</b>	<b>2.0%</b>	<b>\$ 1,045,202</b>	<b>1.9%</b>	<b>\$ 1,290,702</b>
Less taxes	(1,018,000)	(0.0%)	(925,000)	(0.0%)	(1,018,000)
Total net income	<b>\$ 189,095</b>	<b>0.0%</b>	<b>\$ 119,202</b>	<b>0.0%</b>	<b>\$ 272,702</b>
<b>Non-cash adjustments:</b>					
Depreciation	\$ 8,100,000	14.0%	\$ 7,050,000	13.0%	\$ 9,650,000
Amortization	3,014	0.0%	2,367	0.0%	3,657
Share-based compensation	1,018	0.0%	1,018	0.0%	1,018
Change in fair value of derivatives	6,522	0.0%	5,885	0.0%	8,242
Foreign currency translation	22,000	0.0%	18,000	0.0%	25,000
Total non-cash adjustments	<b>\$ 16,653,035</b>	<b>30.0%</b>	<b>\$ 15,143,085</b>	<b>28.0%</b>	<b>\$ 21,915,035</b>

PvPDF

CONDENSED CONSOLIDATED STATEMENTS OF OPERATIONS (Unaudited)					
	For the Year Ended December 31, 2013	For the Year Ended December 31, 2012	For the Year Ended December 31, 2011	For the Year Ended December 31, 2010	For the Year Ended December 31, 2009
<b>Revenues:</b>					
Product Sales	\$ 75,517	\$ 75,028	\$ 65,010	\$ 61,293	\$ 59,240
Services	55,714	55,958	60,790	57,082	54,260
Total revenues	<b>\$ 131,231</b>	<b>\$ 130,986</b>	<b>\$ 125,802</b>	<b>\$ 118,375</b>	<b>\$ 113,500</b>
<b>Cost of sales:</b>					
Product	42,589	40,695	35,922	32,852	30,474
Services	30,000	30,000	33,000	30,000	28,000
Total cost of sales	<b>\$ 72,589</b>	<b>\$ 70,695</b>	<b>\$ 68,922</b>	<b>\$ 62,852</b>	<b>\$ 58,474</b>
<b>gross margin</b>	<b>\$ 58,642</b>	<b>\$ 59,291</b>	<b>\$ 56,880</b>	<b>\$ 55,523</b>	<b>\$ 55,026</b>
<b>Operating expenses:</b>					
Selling, general and administrative	3,507	3,768	9,947	10,571	10,571
R&D	1,197	6,445	25,192	25,192	25,192
Total operating expenses	<b>\$ 4,704</b>	<b>\$ 10,213</b>	<b>\$ 35,139</b>	<b>\$ 35,763</b>	<b>\$ 35,763</b>
<b>Operating income:</b>					
Other (revenue) expense, net	30	(20)	10,394	10,478	10,478
Interest expense, net	1,000	1,000	1,000	1,000	1,000
Provision for (income) taxes	1,010	1,830	50,711	58,268	58,268
Net income	<b>\$ 27,144</b>	<b>\$ 27,144</b>	<b>\$ 41,681</b>	<b>\$ 41,070</b>	<b>\$ 41,070</b>
<b>Income per share:</b>					
Basic	\$ 1.47	\$ 1.38	\$ 0.76	\$ 0.37	\$ 0.37
Diluted	\$ 1.46	\$ 1.38	\$ 0.76	\$ 0.37	\$ 0.37
<b>Dividend on participating convertible common stock:</b>					
Basic	10,159,454	10,035,383	7,149,241	7,149,241	7,149,241
Diluted	10,159,454	10,035,383	7,149,241	7,149,241	7,149,241
<b>Total net income:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
Net cash used in financing activities	<b>\$ 30,511</b>	<b>\$ 30,511</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
Net cash used in operations	<b>\$ 44,023</b>	<b>\$ 44,023</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operations:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Capital expenditures:</b>					
Product	\$ 18,000	\$ 17,056	\$ 10,663	\$ 10,663	\$ 10,663
Services	7,650	11,520	28,267	28,267	28,267
Total capital expenditures	<b>\$ 25,650</b>	<b>\$ 28,576</b>	<b>\$ 38,930</b>	<b>\$ 38,930</b>	<b>\$ 38,930</b>
<b>Working capital and Advances:</b>					
Accounts receivable	\$ 3,327	\$ 3,666	\$ 8,842	\$ 8,154	\$ 8,154
Inventories	1,014	1,016	2,016	2,016	2,016
Total working capital and Advances	<b>\$ 4,341</b>	<b>\$ 4,682</b>	<b>\$ 10,858</b>	<b>\$ 10,170</b>	<b>\$ 10,170</b>
<b>Total net cash used in operations:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash used in financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in investing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in operating activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operating activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in investing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in operating activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operating activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in investing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in operating activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operating activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in investing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in operating activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operating activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in investing activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in investing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) financing activities:</b>					
Debt	1,000	1,000	1,000	1,000	1,000
Dividends	22,483	22,706	34,254	34,254	34,254
Purchase of treasury stock	1,000	1,000	1,000	1,000	1,000
Net cash used in financing activities	<b>\$ 24,483</b>	<b>\$ 24,706</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>	<b>\$ 36,254</b>
<b>Total net cash used in financing activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) operating activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026
Japan	5,025	5,706	20,257	20,257	20,257
Net cash used in operating activities	<b>\$ 73,522</b>	<b>\$ 73,522</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Total net cash used in operating activities:</b>					
For the year ended December 31, 2013	<b>\$ 85,692</b>	<b>\$ 85,692</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>	<b>\$ 58,532</b>
<b>Net cash provided by (used in) investing activities:</b>					
Activities	\$ 6,011	\$ 18,708	\$ 44,254	\$ 44,254	\$ 44,254
Dividends	22,483	22,706	34,254	34,254	34,254
Interest	50,026	50,026	50,026	50,026	50,026

PvMuPDF

## Textract

Apple Inc.					
CONSOLIDATED STATEMENT OF OPERATIONS (Unaudited)					
	For the Year Ended September 27, 2014	For the Year Ended September 28, 2013	For the Year Ended September 29, 2012	For the Year Ended September 30, 2011	For the Year Ended October 1, 2010
<b>Net sales</b>					
Product	\$ 203,541	\$ 196,705	\$ 192,476	\$ 186,330	\$ 186,330
Services	59,081	56,185	53,775	50,775	50,775
<b>Total net sales</b>	<b>\$ 262,622</b>	<b>\$ 212,890</b>	<b>\$ 186,251</b>	<b>\$ 186,330</b>	<b>\$ 186,330</b>
<b>Cost of sales</b>					
Product	\$ 142,963	\$ 140,387	\$ 139,182	\$ 134,747	\$ 134,747
Services	36,571	35,764	34,675	32,775	32,775
<b>Total cost of sales</b>	<b>\$ 179,534</b>	<b>\$ 176,151</b>	<b>\$ 173,857</b>	<b>\$ 167,522</b>	<b>\$ 167,522</b>
<b>Gross margin</b>	<b>\$ 83,088</b>	<b>\$ 36,739</b>	<b>\$ 12,394</b>	<b>\$ 18,808</b>	<b>\$ 18,808</b>
<b>Research and development</b>	<b>\$ 7,007</b>	<b>\$ 6,701</b>	<b>\$ 6,765</b>	<b>\$ 6,765</b>	<b>\$ 6,765</b>
Settle legal and other expenses	\$ 1,733	\$ 1,449	\$ 1,393	\$ 1,393	\$ 1,393
<b>Operating income</b>	<b>\$ 64,350</b>	<b>\$ 28,589</b>	<b>\$ 4,232</b>	<b>\$ 10,623</b>	<b>\$ 10,623</b>
<b>Other operating income and expense</b>	<b>29</b>	<b>257</b>	<b>(257)</b>	<b>257</b>	<b>257</b>
<b>Provision for taxes on income</b>	<b>\$ 4,623</b>	<b>\$ 3,835</b>	<b>\$ 3,771</b>	<b>\$ 3,771</b>	<b>\$ 3,771</b>
<b>Net income</b>	<b>\$ 57,986</b>	<b>\$ 25,567</b>	<b>\$ 4,494</b>	<b>\$ 13,105</b>	<b>\$ 13,105</b>
<b>Net income per share</b>					
Basic	\$ 1.67	\$ 1.20	\$ 0.86	\$ 0.86	\$ 0.86
Diluted	\$ 1.66	\$ 1.19	\$ 0.85	\$ 0.85	\$ 0.85
<b>Net cash provided by operating activities</b>	<b>\$ 64,995</b>	<b>\$ 30,375</b>	<b>\$ 4,375</b>	<b>\$ 13,105</b>	<b>\$ 13,105</b>
<b>Net cash used in investing activities</b>	<b>\$ 16,995,151</b>	<b>\$ 15,073,382</b>	<b>\$ 14,415,713</b>	<b>\$ 16,783,219</b>	<b>\$ 16,783,219</b>
<b>Net cash provided by financing activities</b>	<b>\$ 107,799,000</b>	<b>\$ 100,000</b>	<b>\$ 100,000</b>	<b>\$ 100,000</b>	<b>\$ 100,000</b>
<b>Net increase in cash and cash equivalents</b>	<b>\$ 50,709,849</b>	<b>\$ 45,403,382</b>	<b>\$ 44,915,713</b>	<b>\$ 16,883,219</b>	<b>\$ 16,883,219</b>
<b>Net cash and cash equivalents at beginning of period</b>	<b>\$ 1,000</b>	<b>\$ 1,000</b>	<b>\$ 1,000</b>	<b>\$ 1,000</b>	<b>\$ 1,000</b>
<b>Net cash and cash equivalents at end of period</b>	<b>\$ 50,709,849</b>	<b>\$ 45,403,382</b>	<b>\$ 44,915,713</b>	<b>\$ 16,883,219</b>	<b>\$ 16,883,219</b>
<b>Non-cash working capital changes</b>					
Amortization	\$ 1,007	\$ 16,000	\$ 12,000	\$ 12,000	\$ 12,000
Depreciation	22,162	22,792	24,224	24,224	24,224
Interest	1,014	1,014	1,014	1,014	1,014
Other	5,205	5,700	24,267	24,267	24,267
<b>Total non-cash working capital changes</b>	<b>\$ 28,388</b>	<b>\$ 29,730</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>
<b>Total net cash flow</b>	<b>\$ 50,738,137</b>	<b>\$ 45,373,652</b>	<b>\$ 44,577,482</b>	<b>\$ 16,545,219</b>	<b>\$ 16,545,219</b>
<b>Non-cash working capital changes</b>					
Amortization	\$ 1,007	\$ 16,000	\$ 12,000	\$ 12,000	\$ 12,000
Depreciation	22,162	22,792	24,224	24,224	24,224
Interest	1,014	1,014	1,014	1,014	1,014
Other	5,205	5,700	24,267	24,267	24,267
<b>Debt reduction and extinguishment</b>	<b>6,323</b>	<b>6,655</b>	<b>38,261</b>	<b>38,261</b>	<b>38,261</b>
<b>Total non-cash working capital changes</b>	<b>\$ 28,388</b>	<b>\$ 29,730</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>
<b>Total net cash flow</b>	<b>\$ 50,738,137</b>	<b>\$ 45,373,652</b>	<b>\$ 44,577,482</b>	<b>\$ 16,545,219</b>	<b>\$ 16,545,219</b>
<b>Total net cash flow</b>	<b>\$ 50,738,137</b>	<b>\$ 45,373,652</b>	<b>\$ 44,577,482</b>	<b>\$ 16,545,219</b>	<b>\$ 16,545,219</b>
<b>Non-cash working capital changes</b>					
Amortization	\$ 1,007	\$ 16,000	\$ 12,000	\$ 12,000	\$ 12,000
Depreciation	22,162	22,792	24,224	24,224	24,224
Interest	1,014	1,014	1,014	1,014	1,014
Other	5,205	5,700	24,267	24,267	24,267
<b>Debt reduction and extinguishment</b>	<b>6,323</b>	<b>6,655</b>	<b>38,261</b>	<b>38,261</b>	<b>38,261</b>
<b>Total non-cash working capital changes</b>	<b>\$ 28,388</b>	<b>\$ 29,730</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>	<b>\$ 38,231</b>
<b>Total net cash flow</b>	<b>\$ 50,738,137</b>	<b>\$ 45,373,652</b>	<b>\$ 44,577,482</b>	<b>\$ 16,545,219</b>	<b>\$ 16,545,219</b>

PdfMiner



# What's next for RAG: Agents?

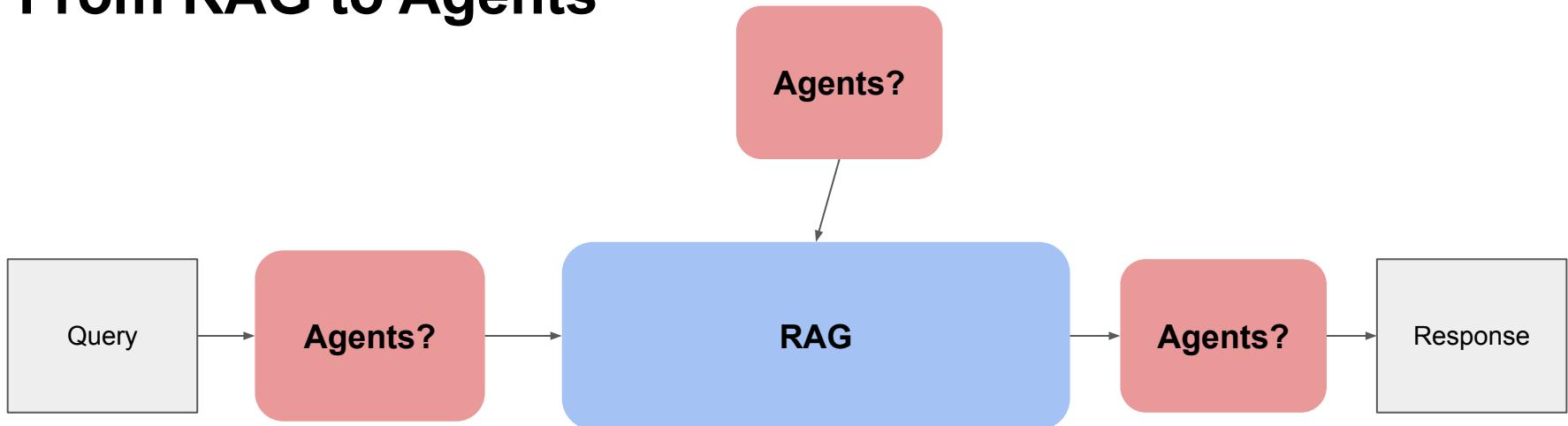


# From RAG to Agents



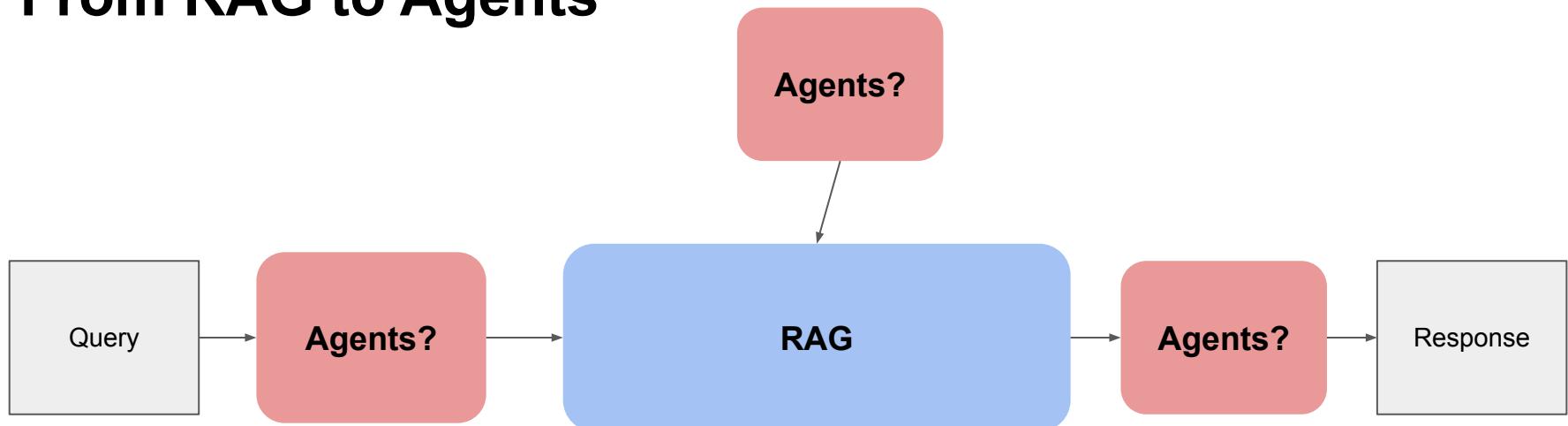


# From RAG to Agents





# From RAG to Agents

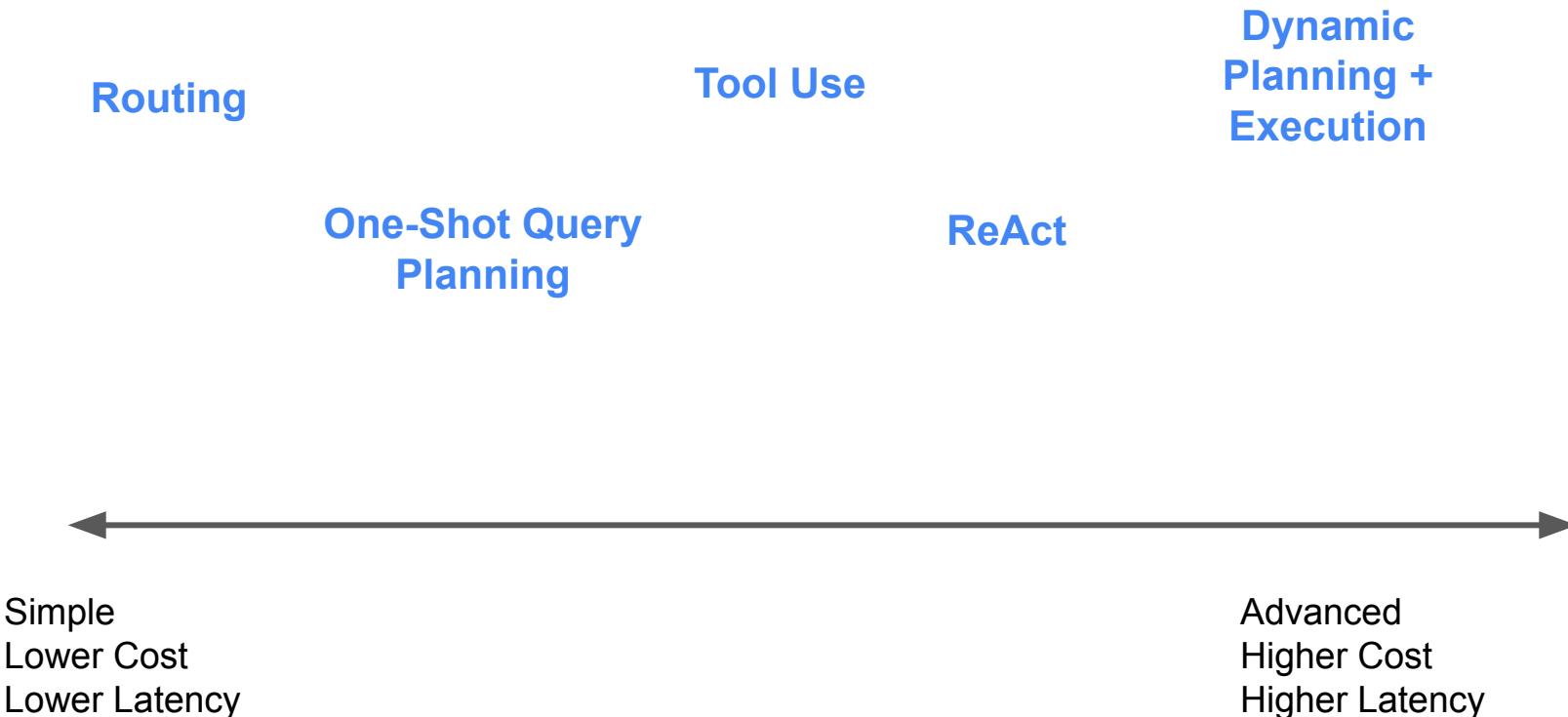


**Agent Definition:** Using LLMs for automated reasoning and tool selection

**RAG is just one Tool:** Agents can decide to use RAG with other tools



# From Simple to Advanced Agents

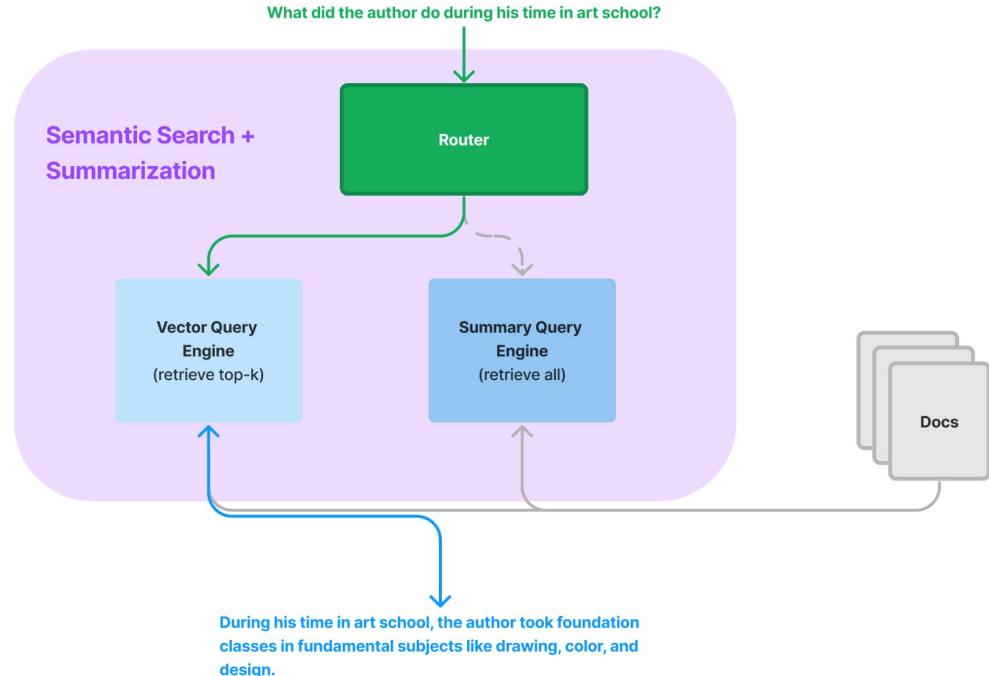




# Routing

Simplest form of agentic reasoning.

Given user query and set of choices, output subset of choices to route query to.

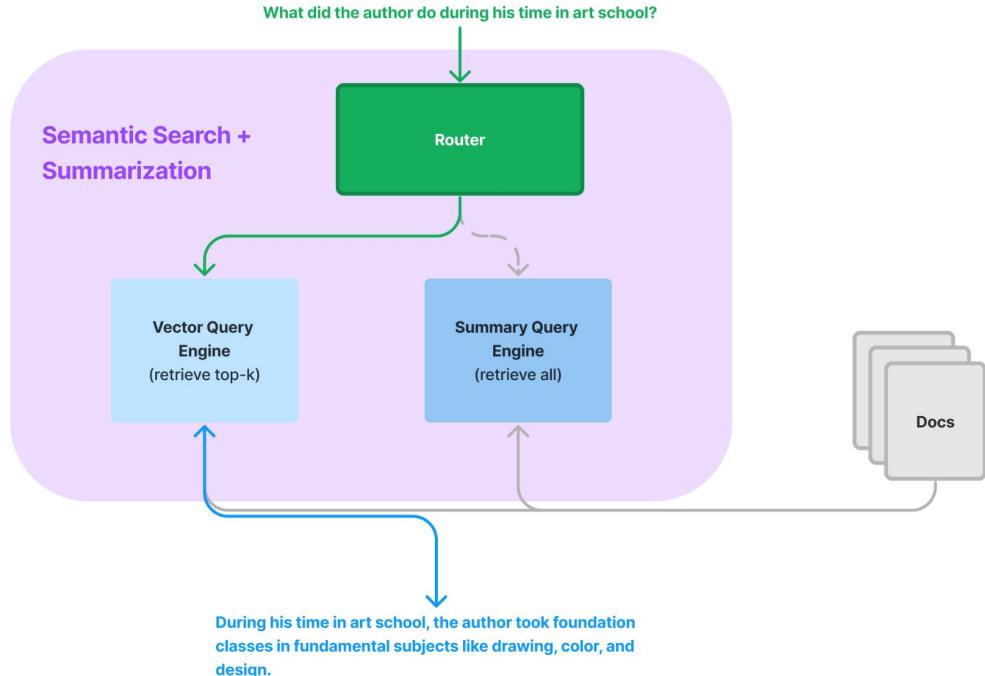




# Routing

**Use Case:** Joint QA and Summarization

[Guide](#)

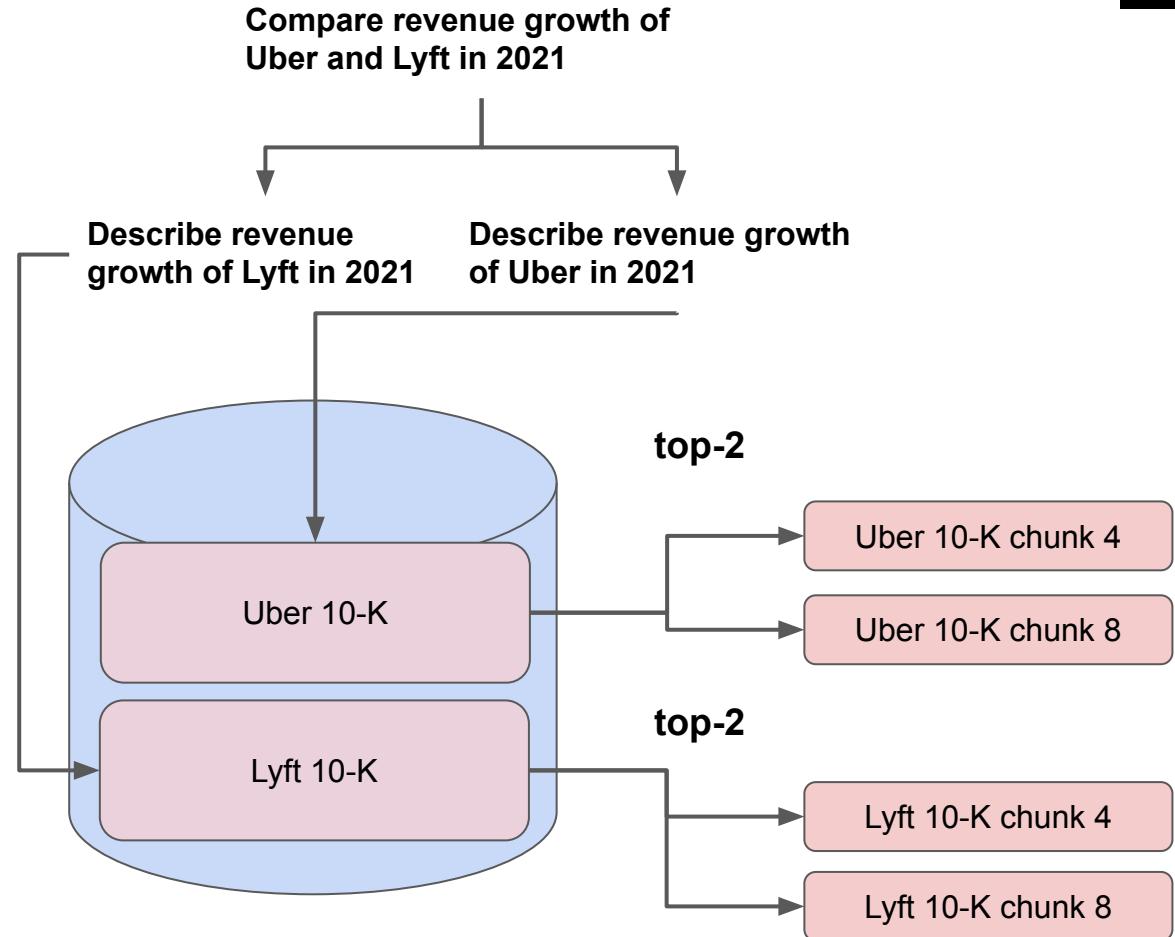




# Query Planning

Break down query into parallelizable sub-queries.

Each sub-query can be executed against any set of RAG pipelines

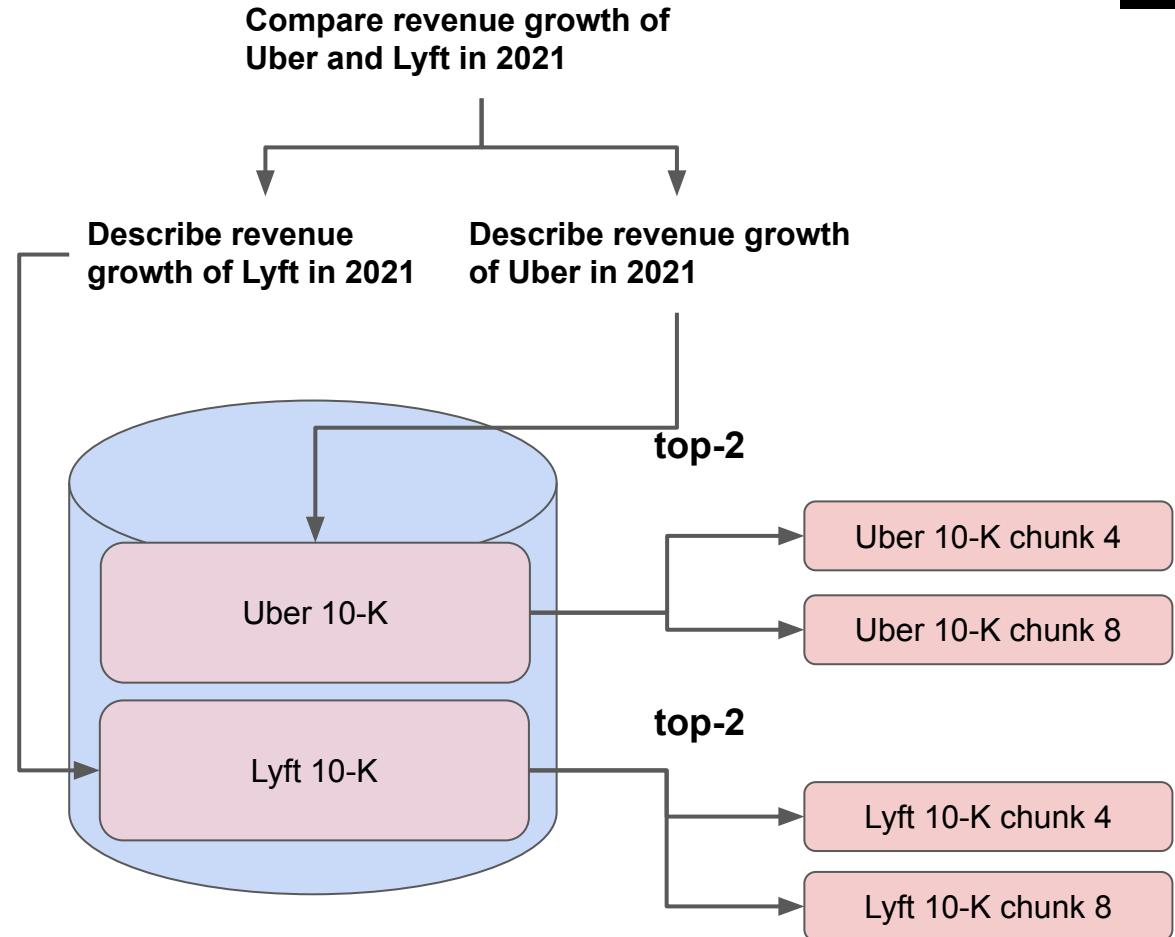




# Query Planning

**Example:** Compare revenue of Uber and Lyft in 2021

[Query Planning Guide](#)



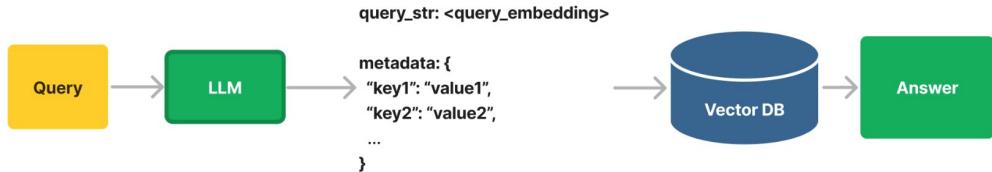


# Tool Use

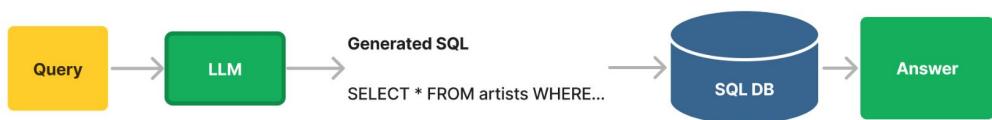
Use an LLM to call an API

Infer the parameters of that API

## Auto-Retrieval



## Text-to-SQL



## Calendar





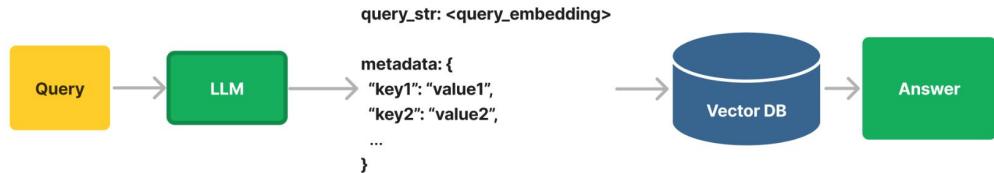
# Tool Use

In normal RAG you just pass through the query.

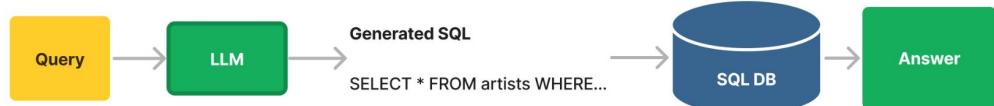
But what if you used the LLM to infer all the parameters for the API interface?

A key capability in many QA use cases (auto-retrieval, text-to-SQL, and more)

## Auto-Retrieval



## Text-to-SQL



## Calendar





# This is cool but

- How can an agent tackle sequential multi-part problems?
- How can an agent maintain state over time?

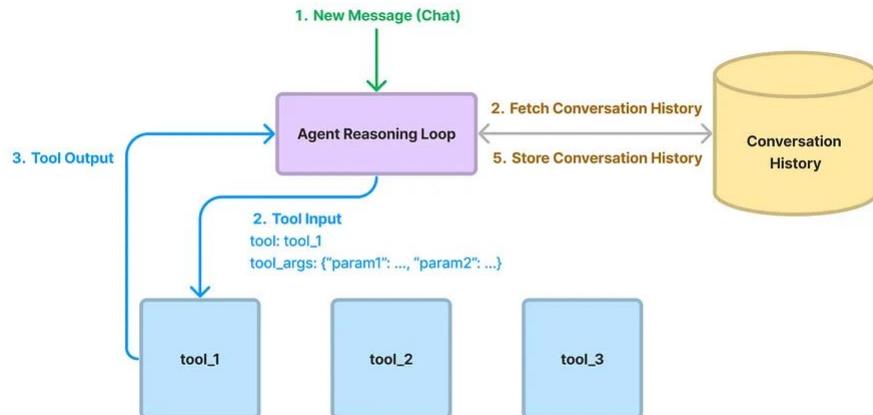


# This is cool but

- How can an agent tackle sequential multi-part problems?
  - Let's make it loop
- How can an agent maintain state over time?
  - Let's add basic memory



# Data Agents - Core Components



## Agent Reasoning Loop

- [ReAct Agent](#) (any LLM)
- [OpenAI Agent](#) (only OAI)

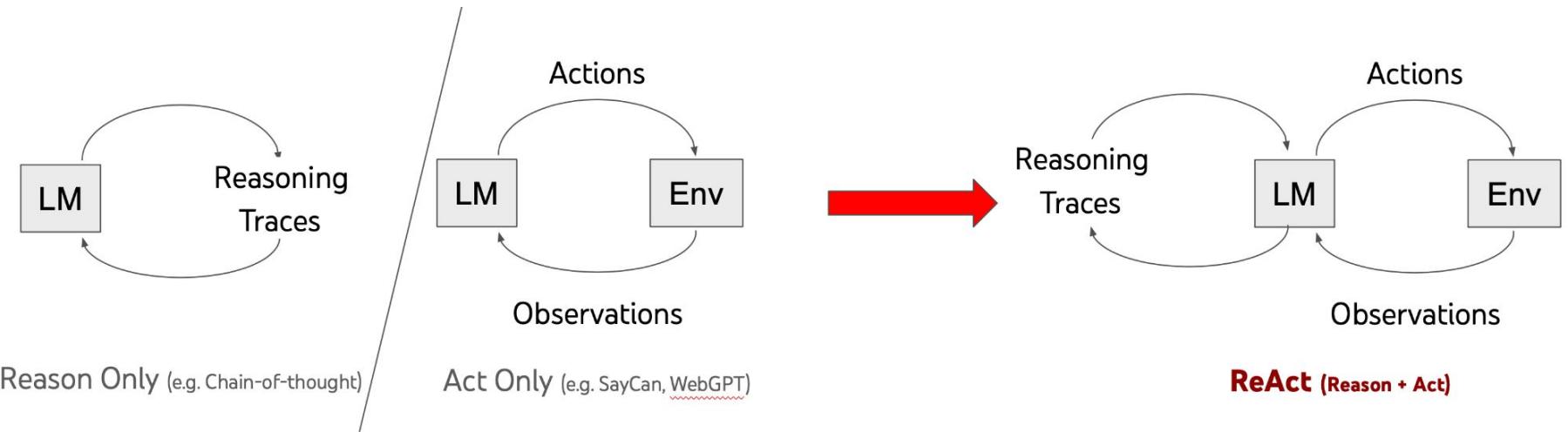
## Tools

[Query Engine Tools \(RAG pipeline\)](#)

[LlamaHub Tools](#) (30+ tools to external services)



# ReAct: Reasoning + Acting with LLMs



Source: <https://react-lm.github.io/>



# ReAct: Reasoning + Acting with LLMs

Add a loop around  
query  
decomposition + tool  
use

```
[]: response = agent.chat(
 "Tell me about the demographics of Houston, and compare that with the demographics of Chicago"
)

Added user message to memory: Tell me about the demographics of Houston, and compare that with the demographics o
f Chicago
==> Calling Function ==
Calling function: vector_tool_Houston with args: {
 "input": "demographics"
}
Got output: Houston has a population of 2,304,580 according to the 2020 U.S. census. In 2017, the estimated popul
ation was 2,312,717, and in 2018 it was 2,325,502. The city has a diverse demographic makeup, with a significant
number of undocumented immigrants residing in the Houston area, comprising nearly 9% of the city's metropolitan p
opulation in 2017. The age distribution in Houston includes a significant number of individuals under 15 and betw
een the ages of 20 to 34. The median age of the city is 33.4. The city has a mix of homeowners and renters, with
an estimated 42.3% of Houstonians owning housing units. The median household income in 2019 was $52,338, and 20.
1% of Houstonians lived at or below the poverty line.
=====

==> Calling Function ==
Calling function: vector_tool_Chicago with args: {
 "input": "demographics"
}
Got output: Chicago experienced rapid population growth during its first hundred years, becoming one of the faste
st-growing cities in the world. From its founding in 1833 with fewer than 200 people, the population grew to over
4,000 within seven years. By 1890, the population had surpassed 1 million, making Chicago the fifth-largest city
in the world at the time. The city's population continued to grow, reaching its highest recorded population of 3.
6 million in 1950. However, in the latter half of the 20th century, Chicago's population declined, dropping to un
der 2.7 million by 2010. The city experienced a rise in population for the 2000 census, followed by a decrease in
2010, and then another increase for the 2020 census. According to U.S. census estimates as of July 2019, the larg
est racial or ethnic groups in Chicago are non-Hispanic White (32.8%), Blacks (30.1%), and Hispanics (29.0%). Add
itionally, Chicago has the third-largest LGBTQ population in the United States, with an estimated 7.5% of the adul
t population identifying as LGBTQ in 2018.
=====
```



# ReAct: Reasoning + Acting with LLMs

Superset of query planning + routing capabilities.

[ReAct + RAG Guide](#)



```
[]: response = agent.chat(
 "Tell me about the demographics of Houston, and compare that with the demographics of Chicago"
)

Added user message to memory: Tell me about the demographics of Houston, and compare that with the demographics o
f Chicago
==> Calling Function ==
Calling function: vector_tool_Houston with args: {
 "input": "demographics"
}
Got output: Houston has a population of 2,304,580 according to the 2020 U.S. census. In 2017, the estimated popul
ation was 2,312,717, and in 2018 it was 2,325,502. The city has a diverse demographic makeup, with a significant
number of undocumented immigrants residing in the Houston area, comprising nearly 9% of the city's metropolitan p
opulation in 2017. The age distribution in Houston includes a significant number of individuals under 15 and betw
een the ages of 20 to 34. The median age of the city is 33.4. The city has a mix of homeowners and renters, with
an estimated 42.3% of Houstonians owning housing units. The median household income in 2019 was $52,338, and 20.
1% of Houstonians lived at or below the poverty line.
=====

==> Calling Function ==
Calling function: vector_tool_Chicago with args: {
 "input": "demographics"
}
Got output: Chicago experienced rapid population growth during its first hundred years, becoming one of the faste
st-growing cities in the world. From its founding in 1833 with fewer than 200 people, the population grew to over
4,000 within seven years. By 1890, the population had surpassed 1 million, making Chicago the fifth-largest city
in the world at the time. The city's population continued to grow, reaching its highest recorded population of 3.
6 million in 1950. However, in the latter half of the 20th century, Chicago's population declined, dropping to un
der 2.7 million by 2010. The city experienced a rise in population for the 2000 census, followed by a decrease in
2010, and then another increase for the 2020 census. According to U.S. census estimates as of July 2019, the larg
est racial or ethnic groups in Chicago are non-Hispanic White (32.8%), Blacks (30.1%), and Hispanics (29.0%). Add
itionally, Chicago has the third-largest LGBTQ population in the United States, with an estimated 7.5% of the adul
t population identifying as LGBTQ in 2018.
=====
```



# Can we make this even better?

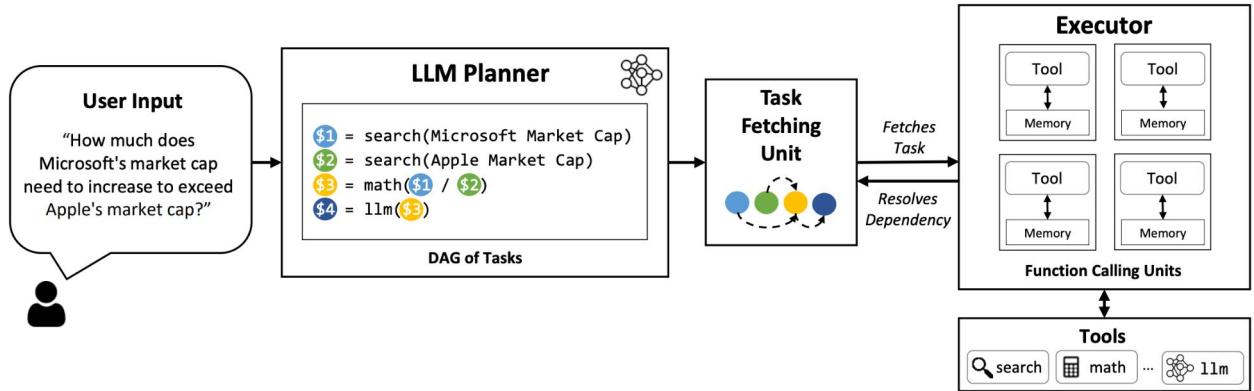
- Stop being so short-sighted - plan ahead at each step
- Parallelize execution where we can



# LLMCompiler

Kim et al. 2023

An agent compiler  
for parallel  
multi-function  
planning +  
execution.



**Figure 2:** Overview of the LLMCompiler framework: the workflow from initial user input to task execution. Beginning with user input, the LLM Planner generates a sequence of tasks with their inter-dependencies. These tasks are then dispatched by the Task Fetching Unit to the Executor based on their dependencies, thus allowing for their parallel executions. For instance, in this example, Task \$1 and \$2 are fetched together for parallel execution of two independent search tasks. After each task is performed, the results (i.e., observations) are forwarded back to the Task Fetching Unit to unblock the dependent tasks after replacing their placeholder variables (e.g., the variable \$1 and \$2 in Task \$3) with actual values. Once all tasks have been executed, the final answer is delivered to the user.



# LLMCompiler

Plan out steps  
beforehand, and  
replan as necessary

## LLMCompiler Agent



```
[17]: response = agent.chat(
 "Is the climate of Chicago or Seattle better during the wintertime?"
)
print(str(response))

> Running step f8fdf4cb-9dde-4aba-996d-edbcee53c4c2 for task 20df27d7-cc27-4311-bb57-b1a6f4ad5799.
> Step count: 0
> Plan: 1. vector_tool_Chicago("climate during wintertime")
2. vector_tool_Seattle("climate during wintertime")
3. join()<END_OF_PLAN>
Ran task: vector_tool_Seattle. Observation: During wintertime, Seattle experiences cool, wet conditions. Extreme cold temperatures, below about 15 °F or -9 °C, are rare due to the moderating influence of the adjacent Puget Sound, the greater Pacific Ocean, and Lake Washington. The city is often cloudy due to frequent storms and lows moving in from the Pacific Ocean, and it has many "rain days". However, the rainfall is often a light drizzle.
Ran task: vector_tool_Chicago. Observation: During wintertime, the city experiences relatively cold and snowy conditions. Blizzards can occur, as they did in winter 2011. The normal winter high from December through March is about 36 °F (2 °C). January and February are the coldest months. A polar vortex in January 2019 nearly broke the city's cold record of -27 °F (-33 °C), which was set on January 20, 1985. Measurable snowfall can continue through the first or second week of April. The city's proximity to Lake Michigan tends to keep the lakefront somewhat cooler in summer and less brutally cold in winter than inland parts of the city and suburbs away from the lake. Northeast winds from wintertime cyclones departing south of the region sometimes bring the city lake-effect snow.
Ran task: join. Observation: None
> Thought: Comparing the two climates, Seattle seems to have a milder winter climate than Chicago.
> Answer: Seattle
Seattle
```



# Tree-based Planning

Tree of Thoughts  
(Yao et al. 2023)

Reasoning via  
Planning (Hao et al.  
2023)

Language Agent  
Tree Search (Zhou  
et al. 2023)

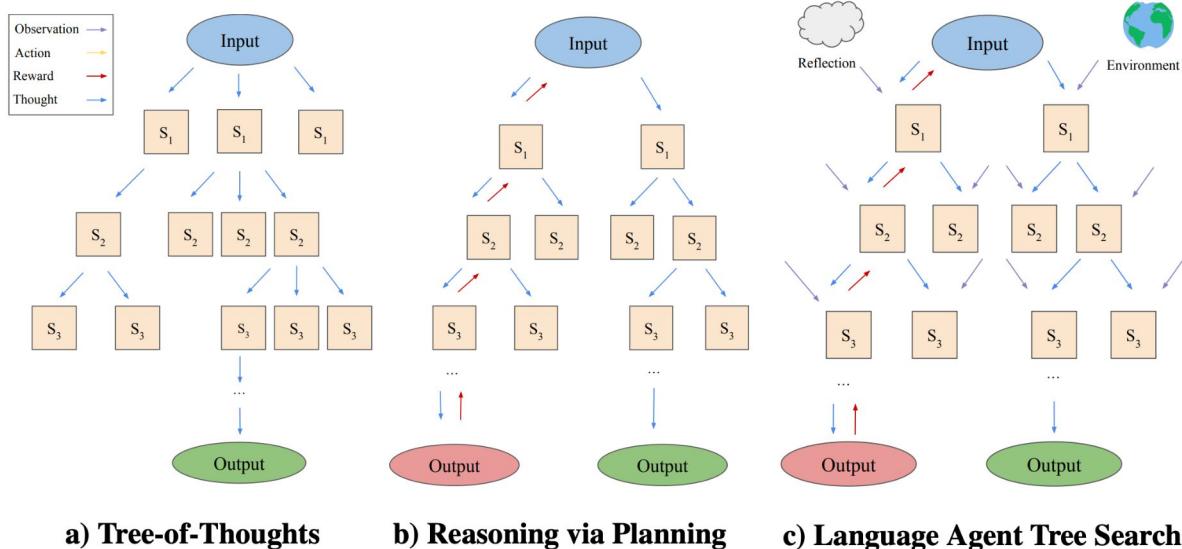


Figure 2: An overview of the differences between LATS and recently proposed LM search algorithms ToT (Yao et al., 2023a) and RAP (Hao et al., 2023). LATS leverages environmental feedback and self-reflection to further adapt search and improve performance.



# Additional Requirements

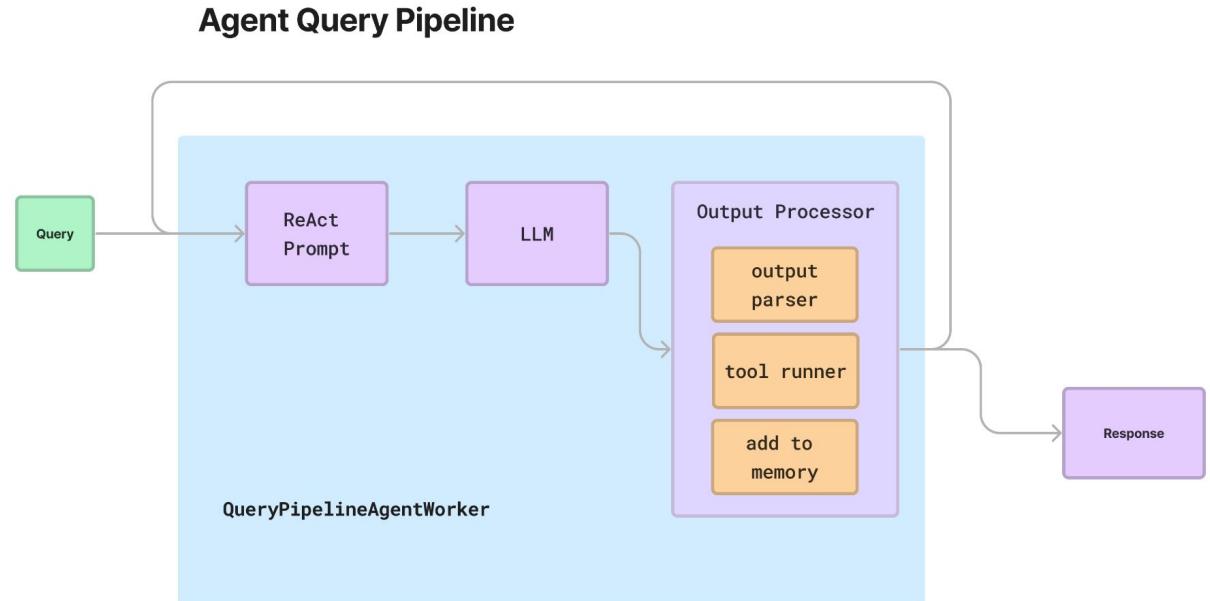
- **Observability:** see the full trace of the agent
  - [Observability Guide](#)
- **Control:** Be able to guide the intermediate steps of an agent *step-by-step*
  - [Lower-Level Agent API](#)
- **Customizability:** Define your own agentic logic around any set of tools.
  - [Custom Agent Guide](#)
  - [Custom Agent with Query Pipeline Guide](#)



# Additional Requirements

Possible through our query pipeline syntax

## Query Pipeline Guide





# What's next for RAG: Long Contexts?



# Is RAG Dead?

Gemini 1.5 Pro has a 1-10M context window.

What does this mean for RAG?

[https://x.com/Francis\\_YAO\\_/status/1759962812229800012?s=20](https://x.com/Francis_YAO_/status/1759962812229800012?s=20)

 **Yao Fu**   
@Francis\_YAO\_ 

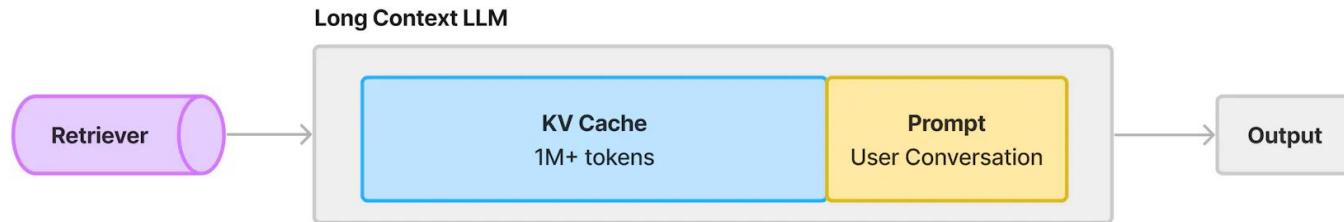
Over the last two days after my claim "long context will replace RAG", I have received quite a few criticisms (thanks and really appreciated!) and many of them stand a reasonable point. Here I have gathered the major counterargument, and try to address them one-by-one (feels like a paper rebuttal):

- **RAG is cheap, long context is expensive.** True, but remember, compared to LLM, BERT-small is also cheap, and n-gram is even cheaper, but they are not used today, because we want the model to be smart first, then makes smart models cheaper -- history of AI tells **it is much easier to make smart models cheaper than making cheap model smart** -- when it is cheap, it's never smart.
- **Long context can mix retrieval and reasoning during the whole decoding processing.** RAG only does the retrieval at the very beginning. Typically, given a question, RAG retrieves the paragraphs that are related to the question, then generates. Long-context does the retrieval for every layer and every token. In many cases the model needs to **do on-the-fly per-token interleaved retrieval and reasoning**, and only knows what to retrieve after getting the results of the first reasoning step. Only long-context can do such cases.
- **RAG supports trillion level tokens, long-context is 1M.** True, but there is a natural distribution of the input document, and I tend to believe **most of the cases that requires retrieval is under million level**. For example, imagine a layer working on a case whose input is related legal documents, or a student learning machine learning whose input are three ML books -- does not feel as long as 1B right?



# Our Position

1. Frameworks are valuable whether or not RAG lives or dies
2. Certain RAG concepts will go away, but others will remain and evolve





# Long Context LLMs will Solve the Following

1. Developers will worry less about tuning chunking algorithms
2. Developers will need to spend less time tuning retrieval and chain-of-thought over single documents
3. Summarization will be easier
4. Personalized memory will be better and easier to build



# Some Challenges Remain

1. 10M tokens is not enough for large document corpuses (hundreds of MB, GB)
2. Embedding models are lagging behind in context length
3. Cost and Latency
4. A KV Cache takes up a significant amount of GPU memory, and has sequential dependencies



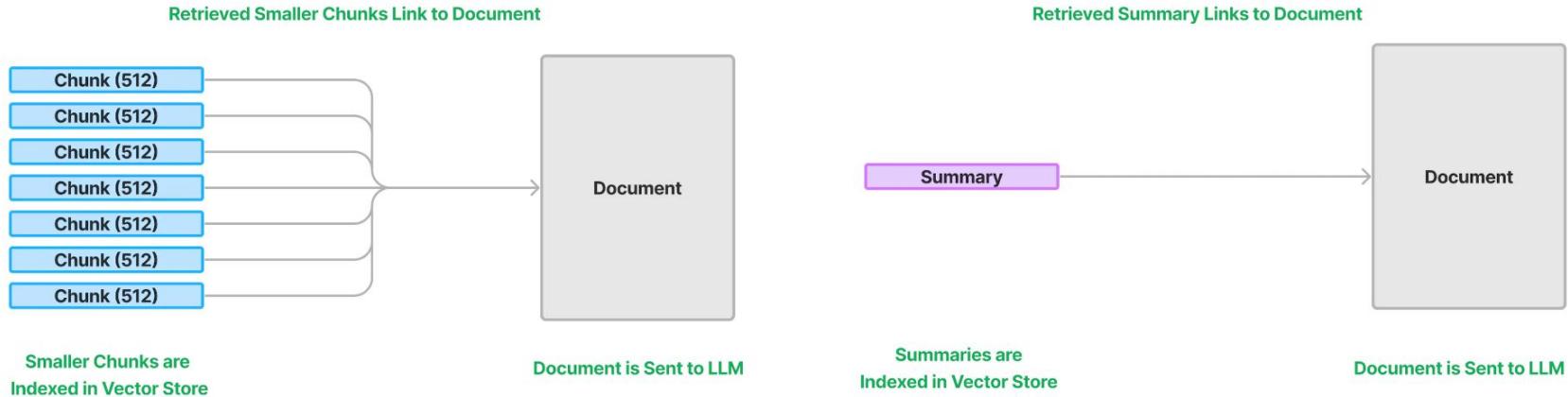
# New RAG Architectures

1. Small to Big Retrieval over Documents
2. Intelligent Routing for Latency/Cost Tradeoffs
3. Retrieval Augmented KV Caching



# Small to Big Retrieval over Documents

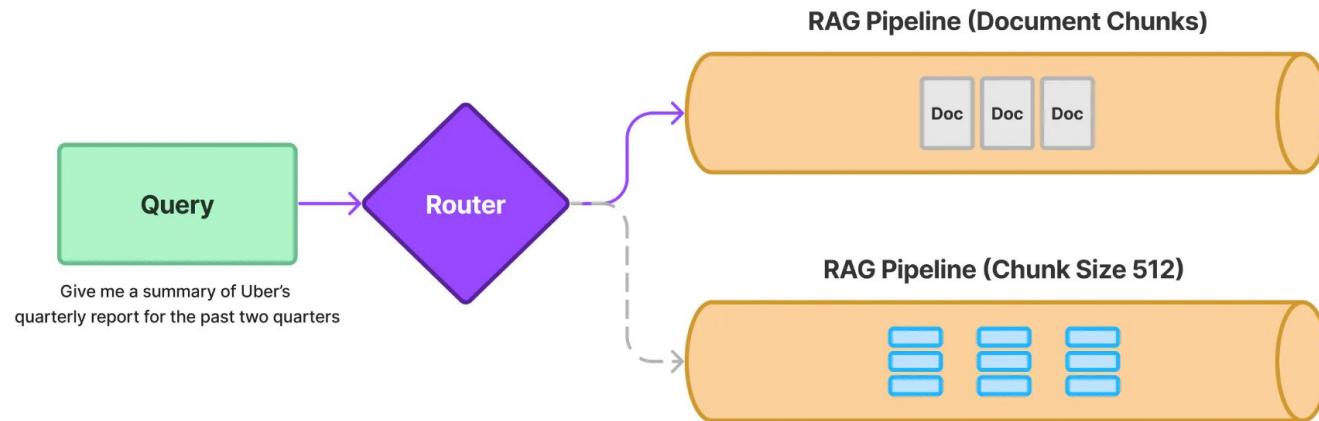
## Small-to-Big Retrieval over Documents





# Intelligent Routing for Latency/Cost Tradeoffs

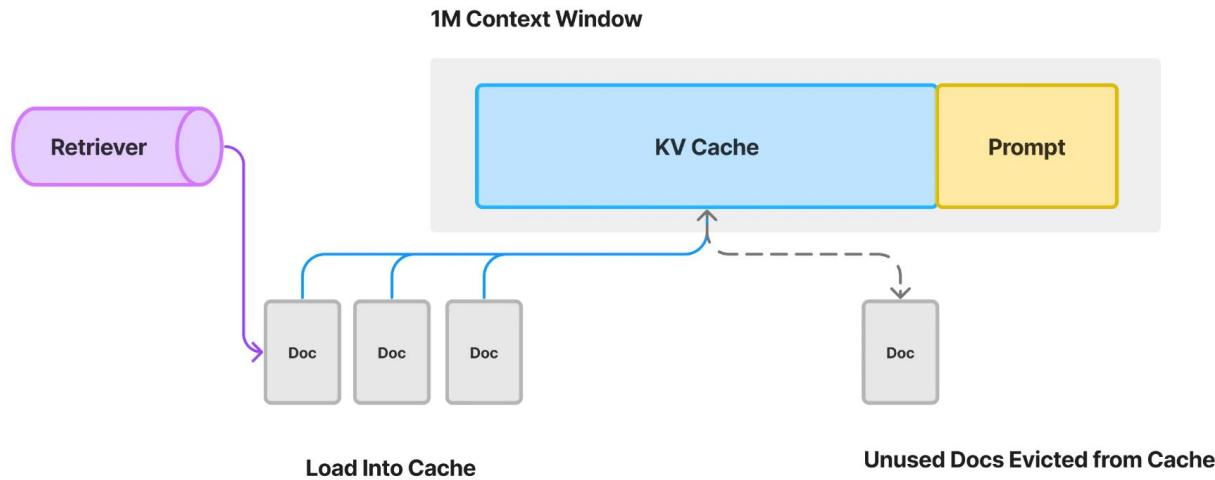
## Intelligent Routing for Latency/Cost Tradeoffs





# Retrieval Augmented KV Caching

## Retrieval for KV Caching





# Thanks for your attention!



## Hiring!

- AI Engineer
- Applied AI Engineer (customer-facing)
- Backend Engineer
- Frontend/Product Engineer
- Typescript Engineer (for LlamaIndex.TS)