



LLM Application Development with LangChain

Lance Martin, LangChain

Upcoming Events



WEBINAR

Build hallucination-free semantic search applications with OriginTrail and Milvus

Branimir Rakic

Founder and CTO, Trace Labs

September 07, 2023 | 9:00 AM PT



WEBINAR

Unlocking Advanced RAG: Citations & Attributions

Yujian Tang

Developer Advocate, Zilliz

September 21, 2023 | 9:00 AM PT



zilliz.com/event



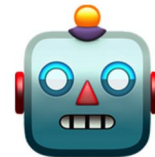
LLM Application Development with LangChain

Lance Martin, LangChain

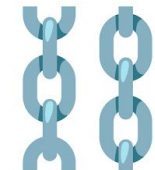
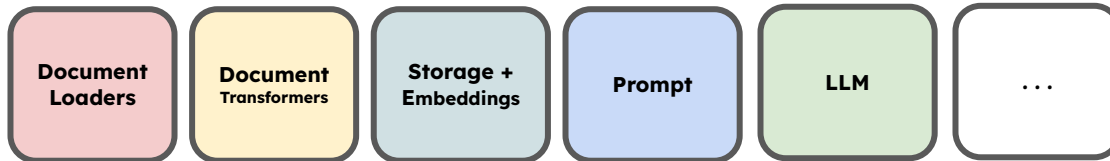
LangChain makes it as easy as possible to develop LLM-powered applications.

Platform (LangSmith) and Library (LangChain)

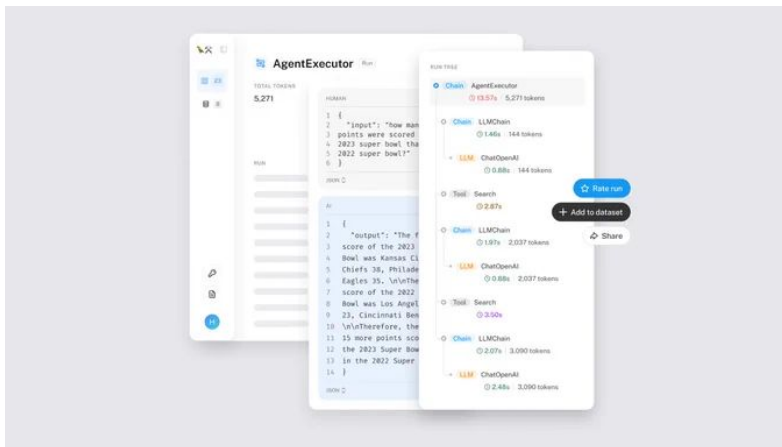
Use-cases
(Applications)



Building Blocks
(Integrations, chains)



Platform
(Debug, Test, Monitor)



Central concepts

Two ways pre-trained LLMs learn

Weight updates via fine-tuning

Cramming before a test

Bad for factual recall, good for tasks

Form (e.g., extraction, text-to-SQL)



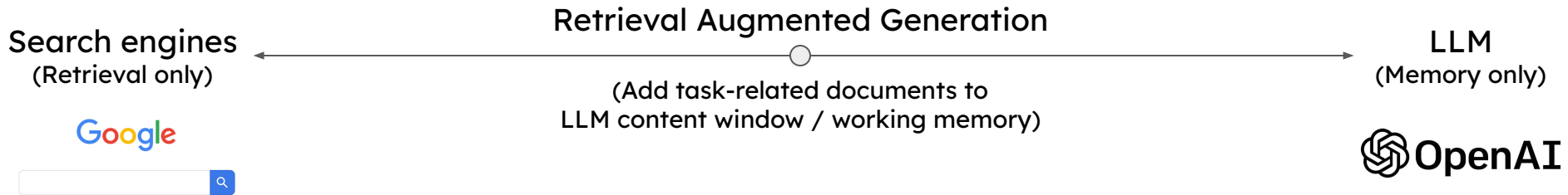
Prompt (e.g., via retrieval)

Open book exam

Good for factual recall

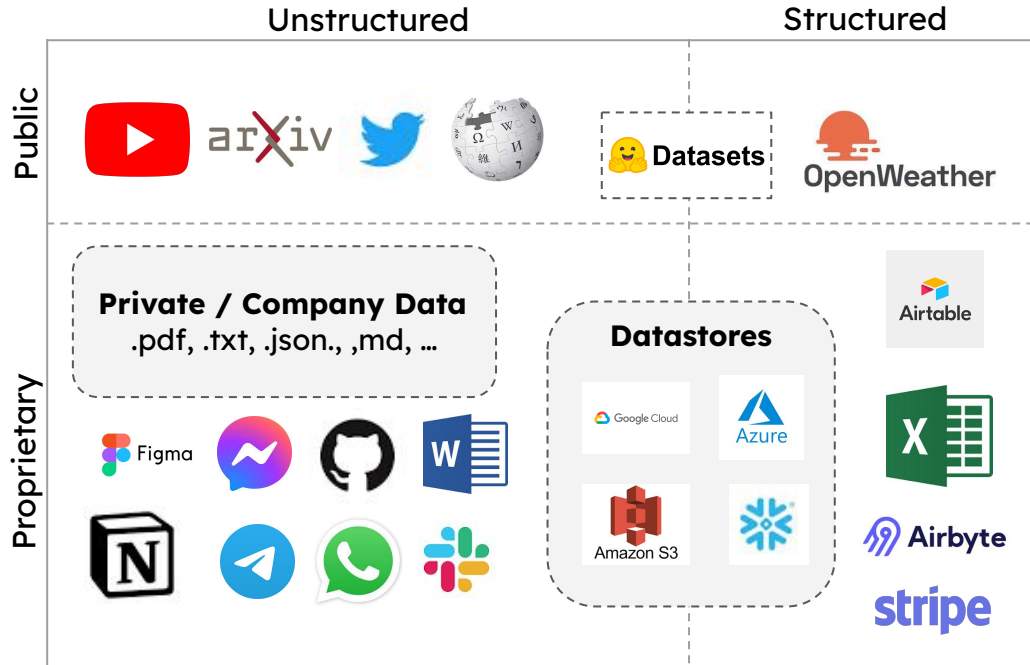
Facts (e.g., QA)



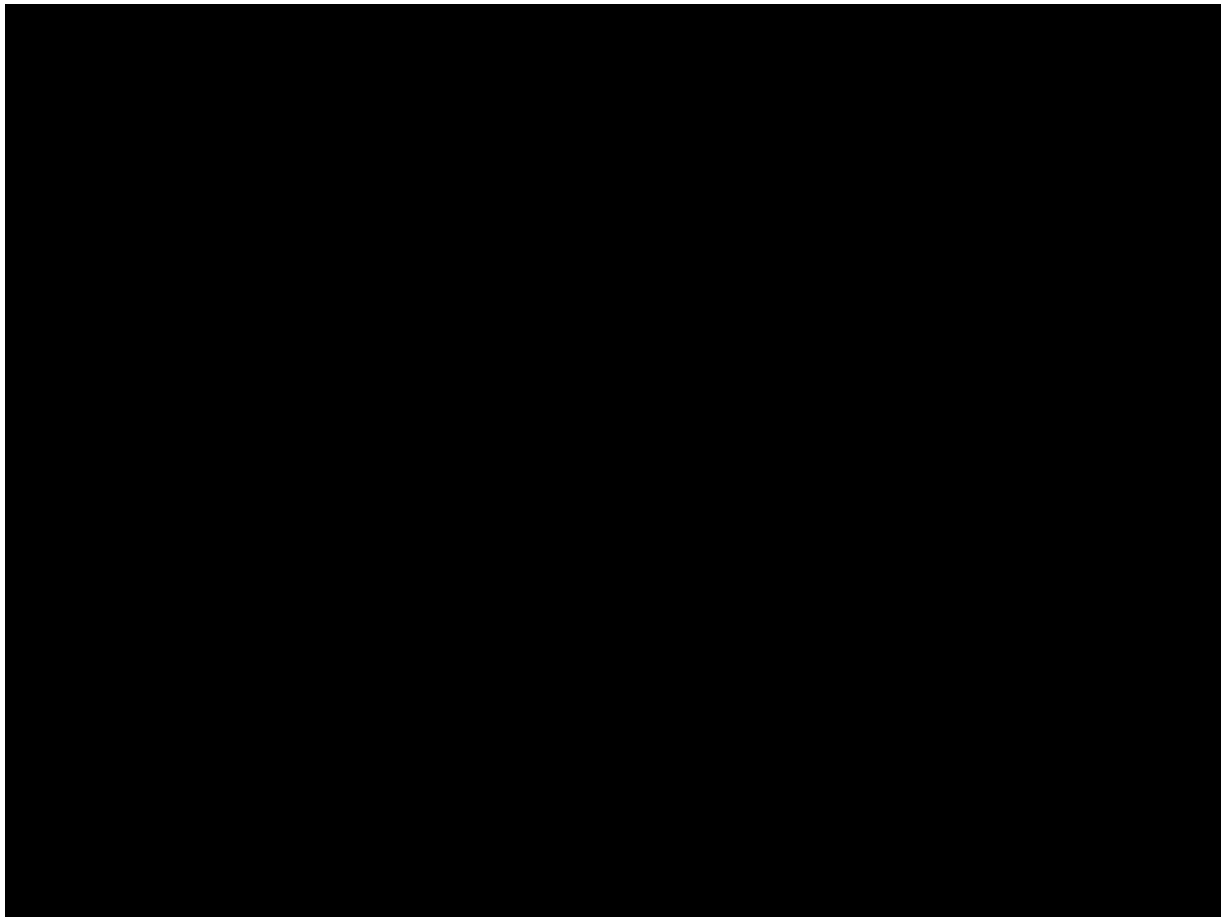


Building Blocks

Document Loaders: > 140 Integrations

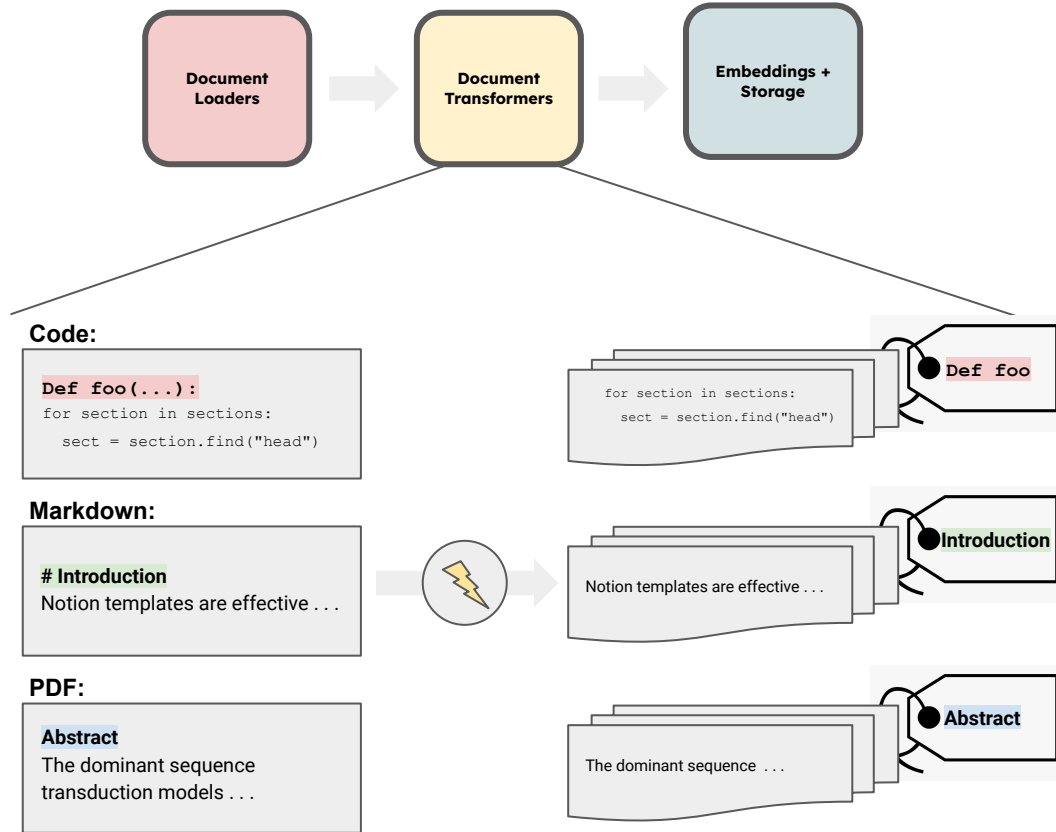


Text splitters

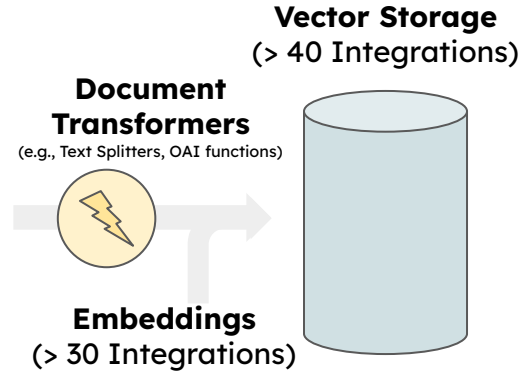


Beyond basic splitting: Context-aware splitting, Function calling (Doctran)

Document Transformers

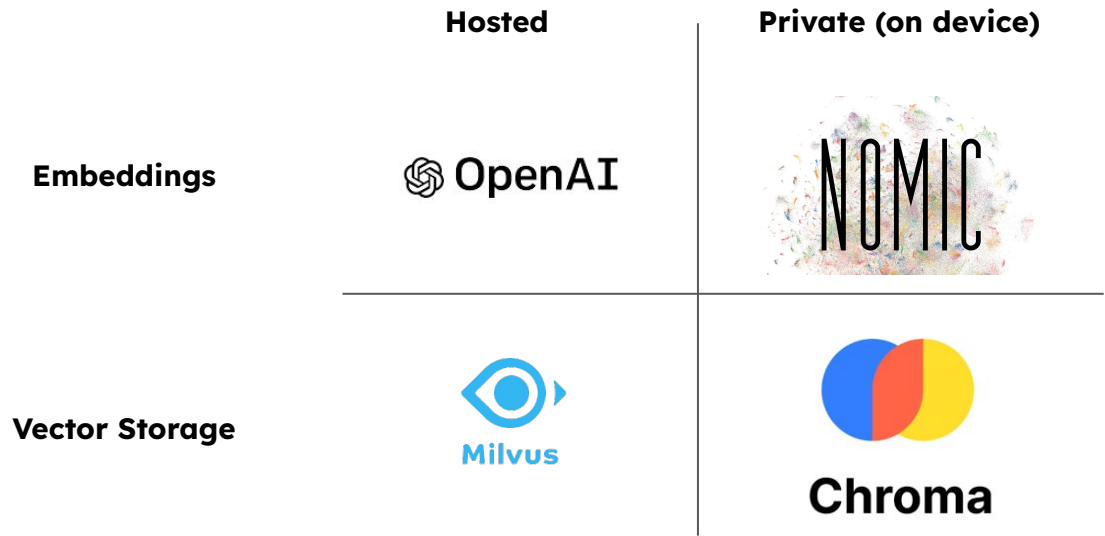


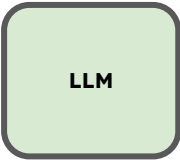
Document Loaders (> 140 Integrations)



Hosted or Private vectorstore + embeddings

Embeddings +
Storage

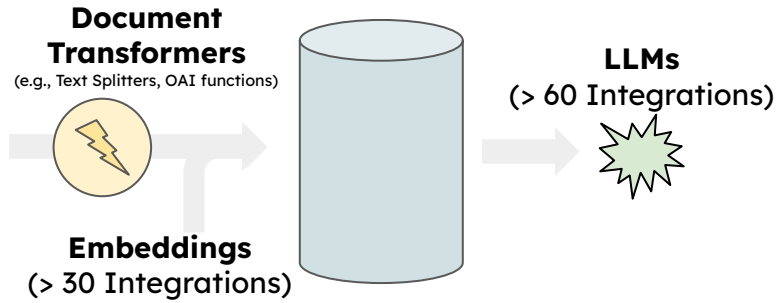




Document Loaders (> 140 Integrations)



Vector Storage (> 40 Integrations)

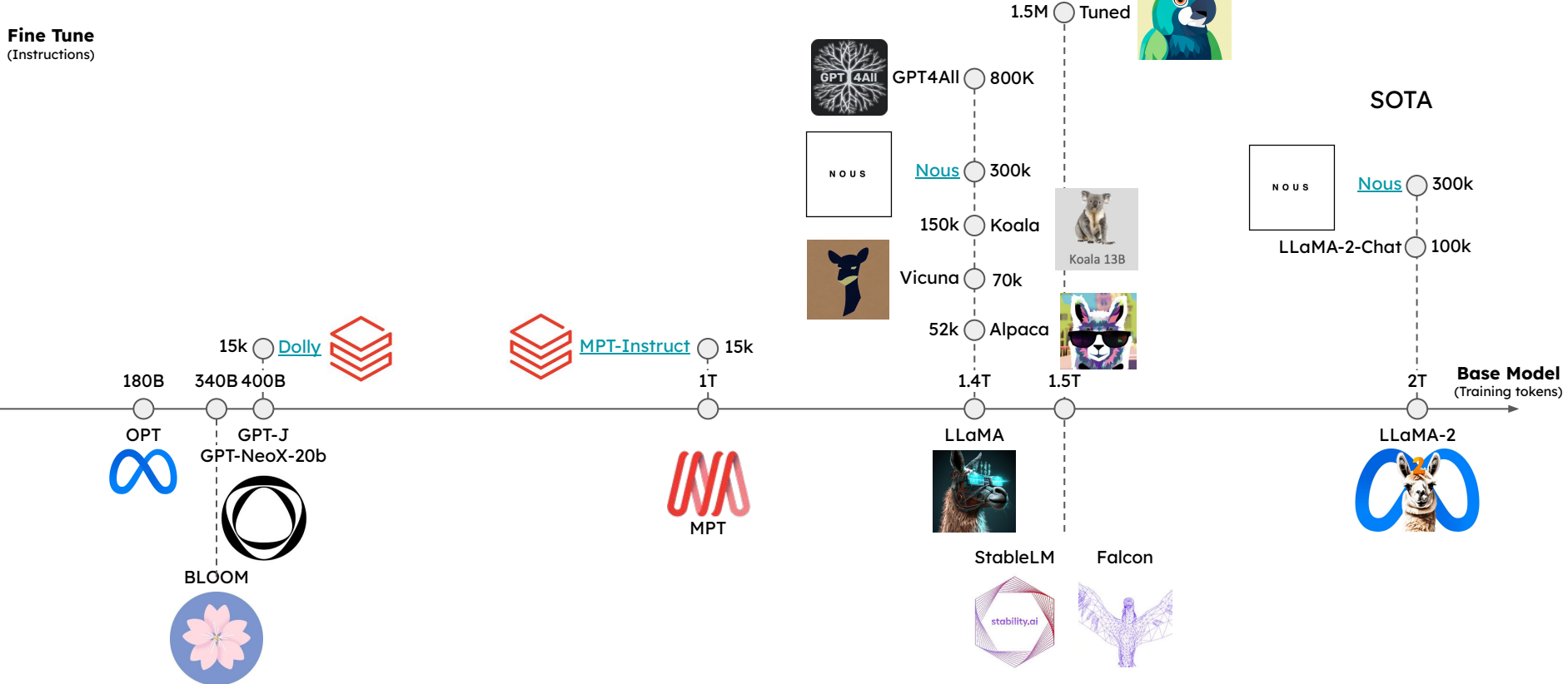
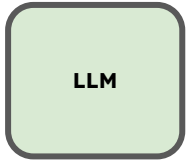


	OpenAI	Anthropic	Llama-2 (SOTA OSS)
Context Window (tokens)	4k - 32k	100k	4k
Performance	GPT-4 SOTA (best overall)	Claude-2 getting closer to GPT4	70b on par w/ GPT-3.5-turbo*
Cost	\$0.06 / 1k token (input)	4-5x cheaper than GPT-4-32K	Free

*Llama2-70b on par w/ GPT-3.5-turbo on [language, but lags on coding](#)

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2	
MMLU (5-shot)	70.0	86.4	69.3	78.3	68.9	Math
TriviaQA (1-shot)	–	–	81.4	86.1	85.0	
Natural Questions (1-shot)	–	–	29.3	37.5	33.0	
GSM8K (8-shot)	57.1	92.0	56.5	80.7	56.8	Language
HumanEval (0-shot)	48.1	67.0	26.2	–	29.9	Code
BIG-Bench Hard (3-shot)	–	–	52.3	65.7	51.2	

Open Source LLMs



OSS models can run on device (private)

LLM

Llama2-13b running ~50 tok / sec (Mac M2 max, 32gb)



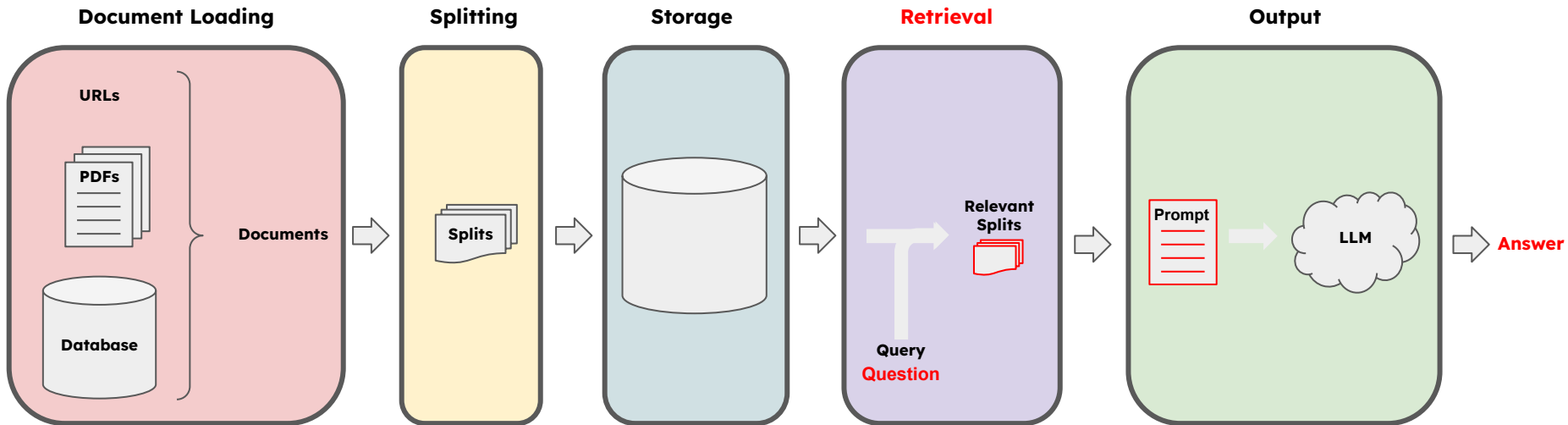


421 integrations

[+ Request an integration](#)Document Loaders 140**Vector Stores 42**Embedding Models 33Chat Models 9LLMs 58Callbacks 24Tools 87Toolkits 16Message Histories 12Popularity ▾

ElasticVectorSearch Wrapper around Elasticsearch as a vector databas... Docs Github ❤️ 130	Qdrant Wrapper around Qdrant vector database. To use... Docs Github ❤️ 111	SupabaseVectorStore VectorStore for a Supabase postgres database.... Docs Github ❤️ 95	Milvus Initialize wrapper around the milvus vector... Docs Github ❤️ 94
Zilliz Initialize wrapper around the Zilliz vector database... Docs Github ❤️ 83	OpenSearchVectorSearch Wrapper around OpenSearch as a vector databas... Docs Github ❤️ 75	Pinecone Wrapper around Pinecone vector database. To us... Docs Github ❤️ 73	Weaviate Wrapper around Weaviate vector database. To us... Docs Github ❤️ 59
LanceDB Wrapper around LanceDB vector database. To use... Docs Github ❤️ 49	Redis Wrapper around Redis vector database. To use, yo... Docs Github ❤️ 45	Marqo Wrapper around Marqo database. Marqo indexes... Docs Github ❤️ 36	DeepLake Wrapper around Deep Lake, a data lake for deep... Docs Github ❤️ 35
AwaDB Interface implemented by AwaDB vector stores. Docs Github ❤️ 33	Cassandra Wrapper around Cassandra embeddings platform... Docs Github ❤️ 30	DocArrayHnswSearch Wrapper around HnswLib storage. To use it, you... Docs Github ❤️ 26	DocArrayInMemorySearch Wrapper around in-memory storage for exact... Docs Github ❤️ 24

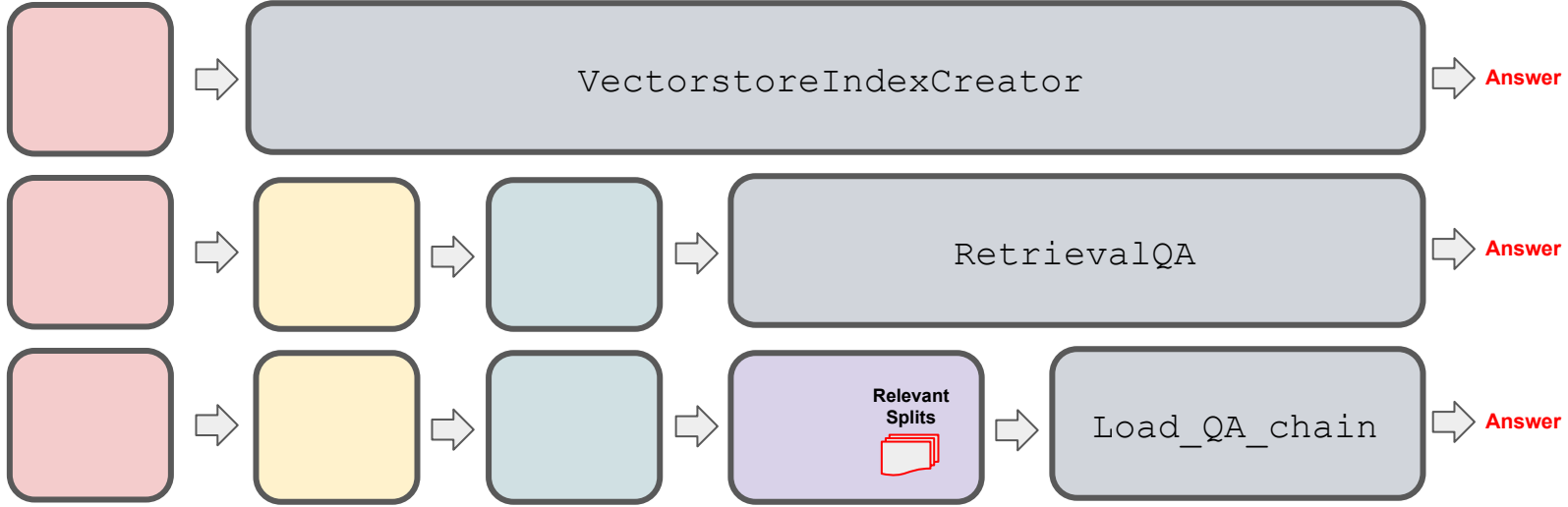
Use Cases



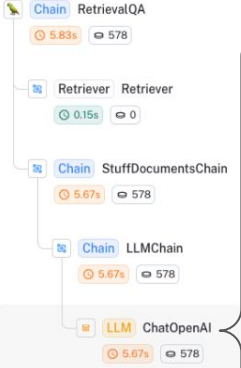
Pick desired level of abstraction

RAG

Abstraction / Simplicity



Trace



ChatOpenAI

Run Feedback Metadata

Share Rate Run Add to Dataset Open in Playground

Prompt

SYSTEM

1 Use the following pieces of context to answer the users question.
2 If you don't know the answer, just say that you don't know, don't try to make up an answer.
3
4 set into a train and a validation split. So, in particular, we're going to take the first 90% of the data set and consider that to be the training data for the transformer. And we're going to withhold the last 10% at the end of it to be the validation data. And this will help us understand to what extent our model is overfitting. So, we're going to basically hide and keep the validation data on the side, because we don't want just a perfect memorization of this exact Shakespeare. We want a
5
6 Now, the batch size here is how many independent sequences we are processing every forward-backward pass of the transformer. The block size, as I explained, is the maximum context length to make those predictions. So, let's say batch size 4, block size 8, and then here's how we get batch for any arbitrary split. If the split is a training split, then we're going to look at train data, otherwise at val data. That gives us the data array. And then, when I generate random positions to grab a chunk
7
8 each other, and so on. So, let me basically just generalize this and introduce a batch dimension. Here's a chunk of code. Let me just run it, and then I'm going to explain what it does. So, here, because we're going to start sampling random locations in the data sets to pull chunks from, I am setting the seed in the random number generator so that the numbers I see here are going to be the same numbers you see later if you try to reproduce this. Now, the batch size here is how many independent
9
10 when I call the set constructor on it, I'm just going to get the set of all the characters that occur in this text. And then I call list on that to create a list of those characters instead of just a set so that I have an ordering, an arbitrary ordering. And then I sort that. So, basically, we get just all the characters that occur in the entire dataset and they're sorted. Now, the number of them is going to be our vocabulary size. These are the possible elements of our sequences. And we see

Retrieved docs

HUMAN

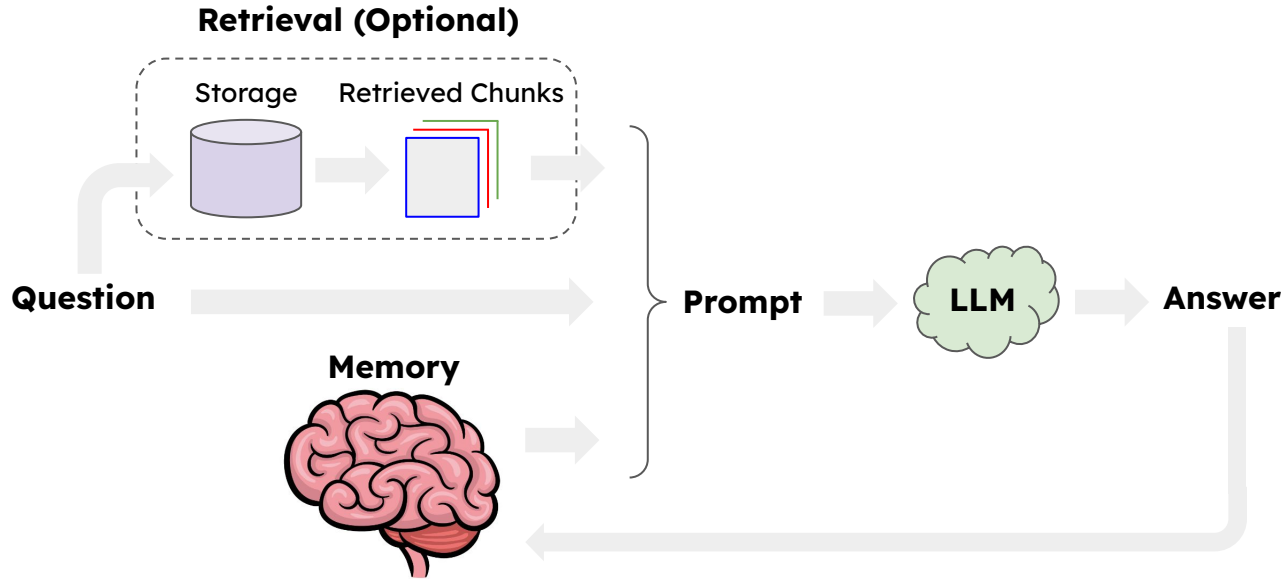
1 What is the purpose of using a validation set?

Question

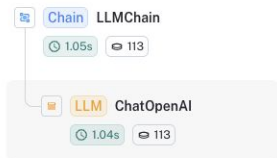
AI

Response

1 The purpose of using a validation set is to understand to what extent the model is overfitting. It is withheld from the training data and kept aside to test the model's performance on unseen data. This helps ensure that the model is not just memorizing the training data but is able to generalize and make accurate predictions on new, unseen data.



Trace



ChatOpenAI

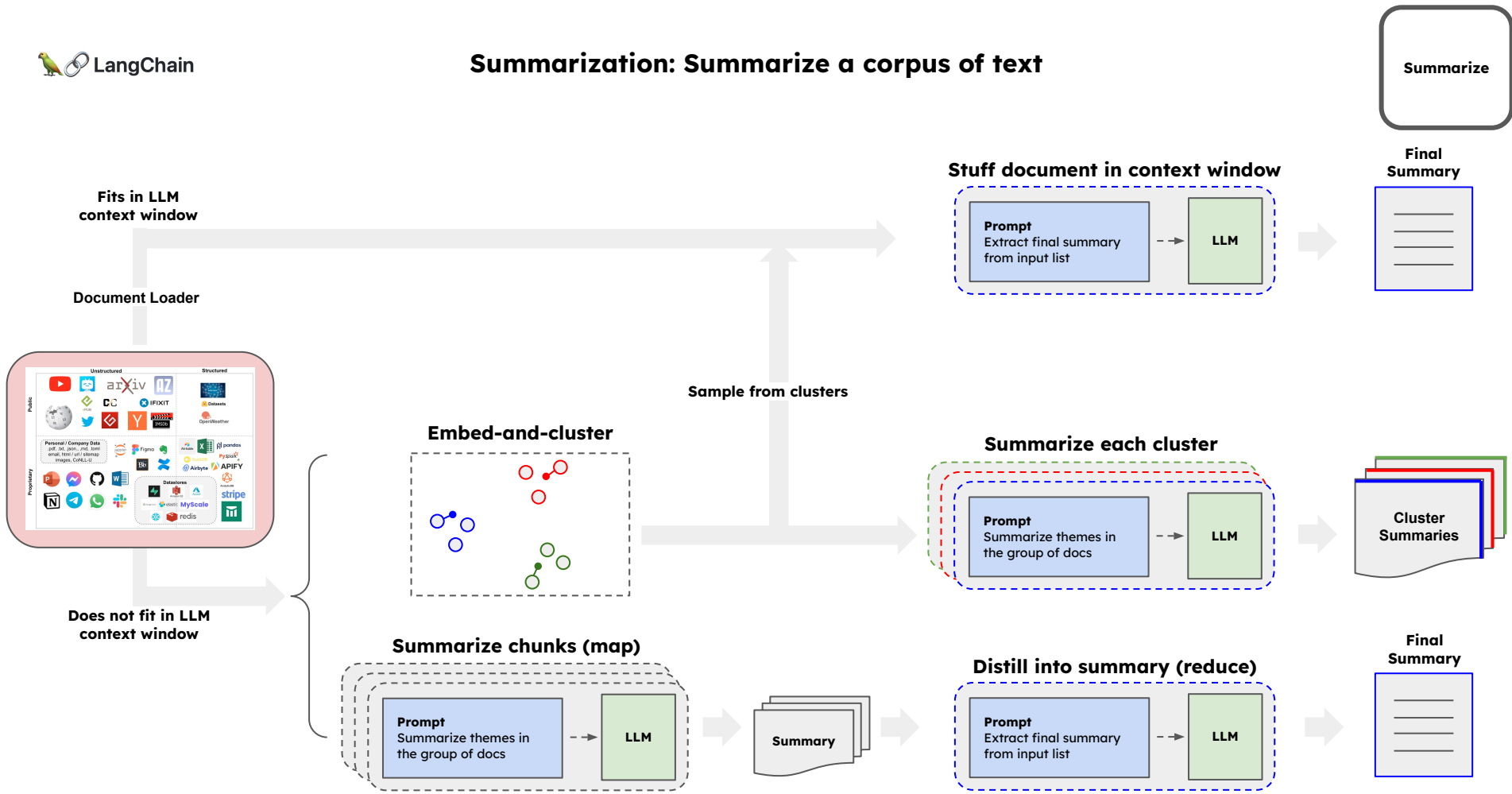
Run Feedback Metadata

Prompt

Role	Message
SYSTEM	
1	You are a nice chatbot having a conversation with a human.
HUMAN	
1	hi
AI	
1	Hello! How can I assist you today?
HUMAN	
1	Translate this sentence from English to French: I love programming.
AI	
1	Sure! The translation of "I love programming" from English to French is "J'adore programmer."
HUMAN	
1	Now translate the sentence to German.
AI	
1	Certainly! The translation of "I love programming" from English to German is "Ich liebe das Programmieren."

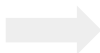
Chat history

Summarization: Summarize a corpus of text



Summarized themes from LangChain questions using different methods and LLMs

User Questions
(Thousands)



	OpenAI	Anthropic	OpenAI
Approach	Map-Reduce	Map-Reduce	Clustering
Model	GPT3.5-16k, GPT4	Claude-2	GPT4
Langchain Usage and Implementation	55%	55%	48%
RAG QA / Chat	N / A	N / A	28%
Prompt Engineering and Customization	10%	15%	7%
Tools and Agents	15%	15%	7%
Vector Stores	N / A	8%	5%
Document Loaders	N / A	8%	N / A
Integration with External Tools and Services	10%	N / A	N / A
Language Models and Embeddings	10%	N / A	5%

Input

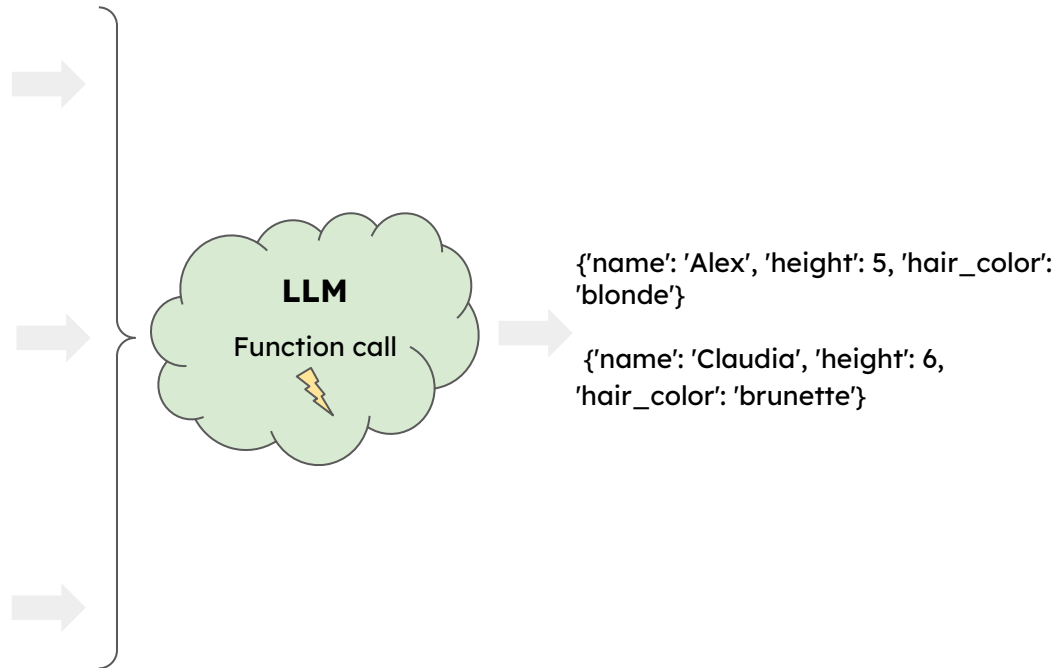
Alex is 5 feet tall. Claudia is 1 feet taller Alex and jumps higher than him. Claudia is a brunette and Alex is blonde.

Schema (tell LLM schema we want)

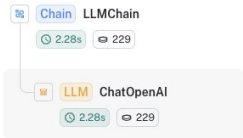
```
schema = {  
  "properties": {  
    "name": {"type": "string"},  
    "height": {"type": "integer"},  
    "hair_color": {"type": "string"},  
  },  
}
```

Function (tell the LLM function)

```
"name": "information_extraction",  
"description": "Extracts information from the passage.",  
"parameters": {schema}
```



Trace



ChatOpenAI

Share

Run Feedback Metadata

HUMAN

Prompt

```
1 Extract and save the relevant entities mentioned in the following passage together with their properties.
2
3 Only extract the properties mentioned in the 'information_extraction' function.
4
5 If a property is not present and is not required in the function parameters, do not include it in the output.
6
7 Passage:
8 Alex is 5 feet tall. Claudia is 1 foot taller than Alex and jumps higher than him. Claudia is a brunette and Alex is blonde.
9
```

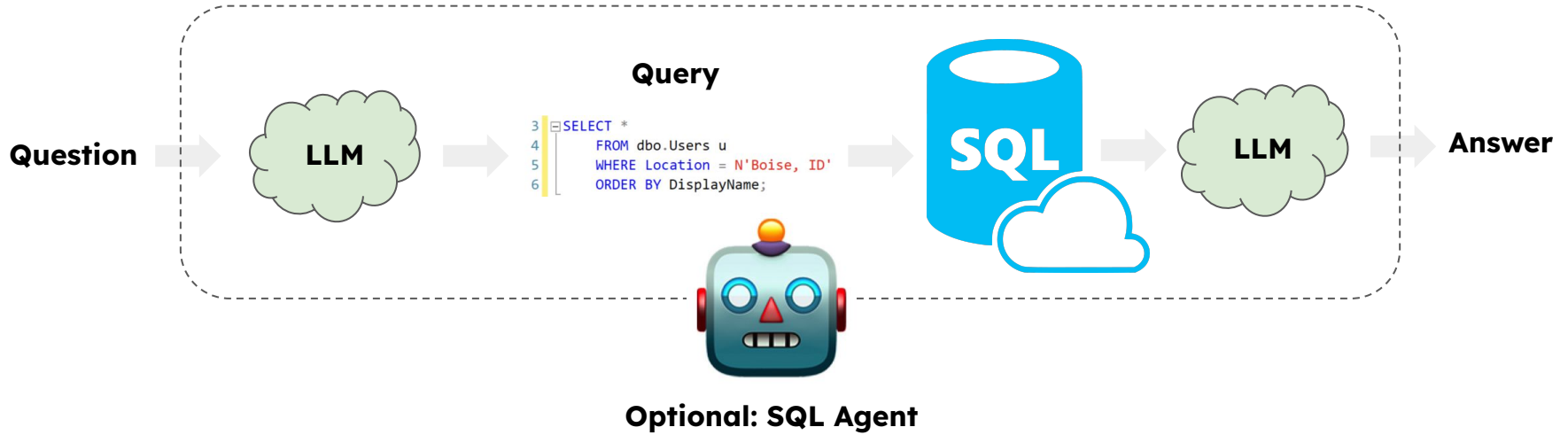
AI

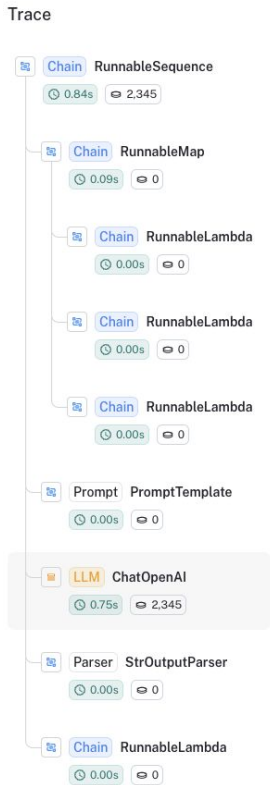
Response

```
1 function_call:
2   name: information_extraction
3   arguments: |-
4     {
5       "info": [
6         {
7           "name": "Alex",
8           "height": 5,
9           "hair_color": "blonde"
10        },
11        {
12          "name": "Claudia",
13          "height": 6,
14          "hair_color": "brunette"
15        }
16      ]
17    }
```

Output from function call → [JSON parser](#)

YAML





ChatOpenAI

Run Feedback Metadata

Prompt

```

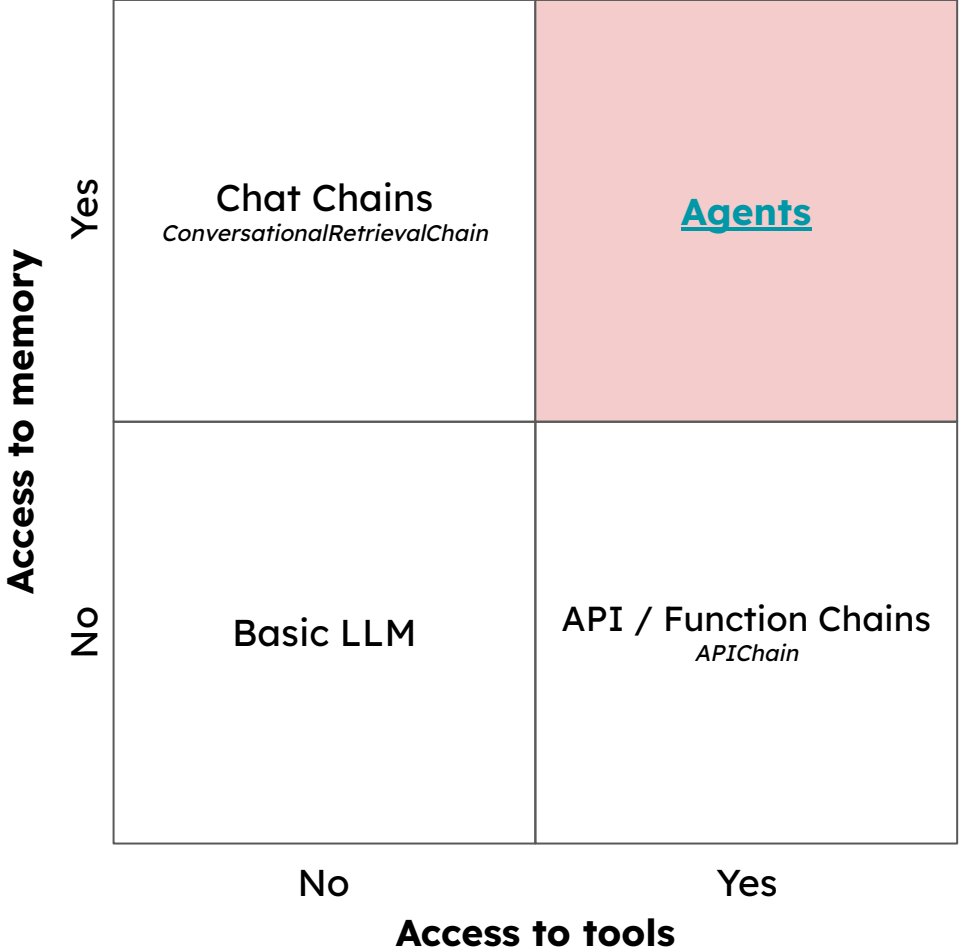
5 Pay attention to use date('now') function to get the current date, if the question involves "today".
6
7 Use the following format:
8
9 Question: Question here
10 SQLQuery: SQL Query to run
11 SQLResult: Result of the SQLQuery
12 Answer: Final answer here
13
14 Only use the following tables:
15
16 CREATE TABLE "Album" (
17     "AlbumId" INTEGER NOT NULL,
18     "Title" NVARCHAR(160) NOT NULL,
19     "ArtistId" INTEGER NOT NULL,
20     PRIMARY KEY ("AlbumId"),
21     FOREIGN KEY("ArtistId") REFERENCES "Artist" ("ArtistId")
22 )
23
24 /*
25 3 rows from Album table:
26 AlbumId Title ArtistId
27 1 For Those About To Rock We Salute You 1
28 2 Balls to the Wall 2
29 3 Restless and Wild 2
30 */
31
          
```

CREATE TABLE description for each table and
and three example rows in a SELECT statement

Response

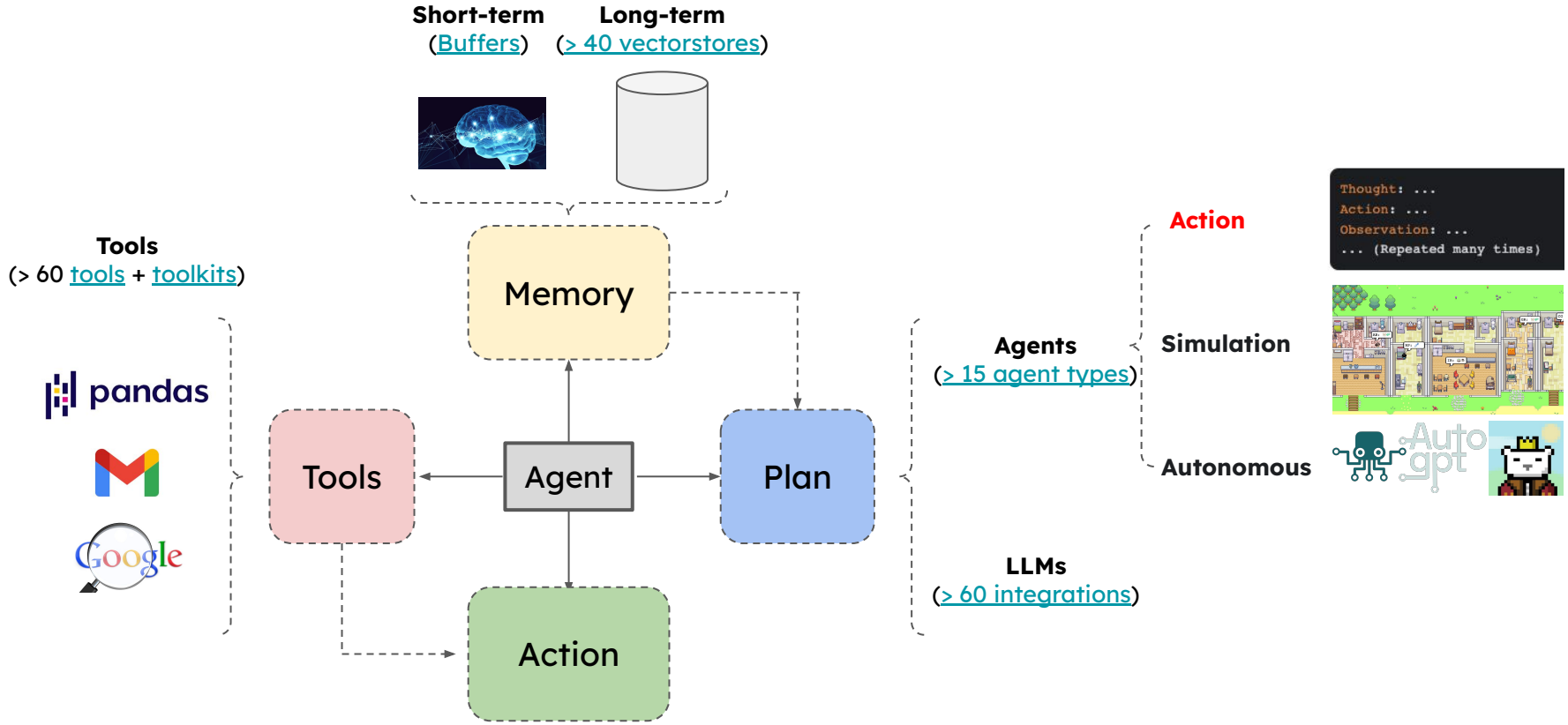
```

1 SELECT COUNT(*) FROM Employee
          
```



Large agent ecosystem (will focus on ReAct as one example)

Agents



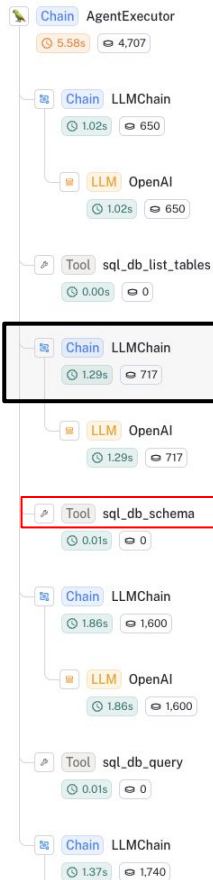
Yes	<u>Say-Can</u>	<u>ReAct</u>
No	<u>Standard Prompting</u>	<u>Chain-of-thought*</u>
	No	Yes

Action-Observation (Tool Use)

Multi-Step Reasoning

*Condition LLM to show its work

Trace



Uses tool at next step

LLMChain

Run Feedback Metadata

Prompt

Copy

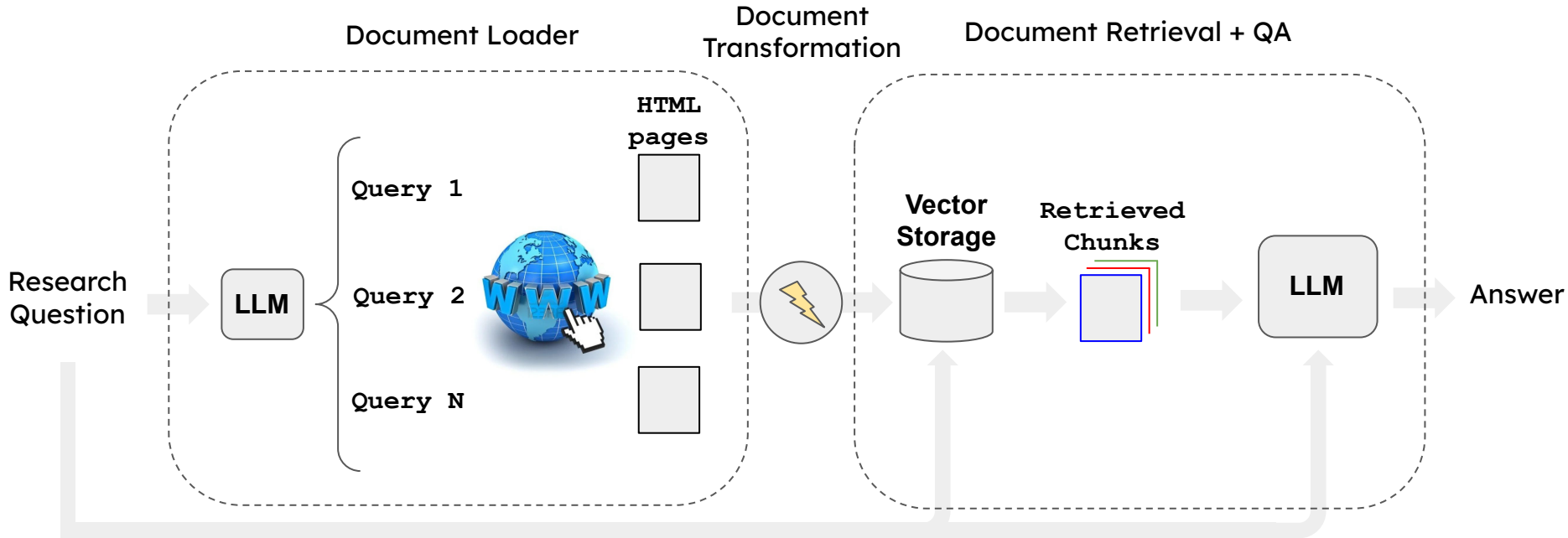
```
1 stop:
2 - |
3
4 Observation:
5 - |
6
7 Observation:
8 input: List the total sales per country. Which country's customers spent the most?
9 agent_scratchpad: |
10 Action: sql_db_list_tables ← Tool / action
11 Action Input:
12 Observation: Album, Artist, Customer, Employee, Genre, Invoice, InvoiceLine, MediaType, Playlist, ← Observation
13 PlaylistTrack, Track
14 Thought:
```

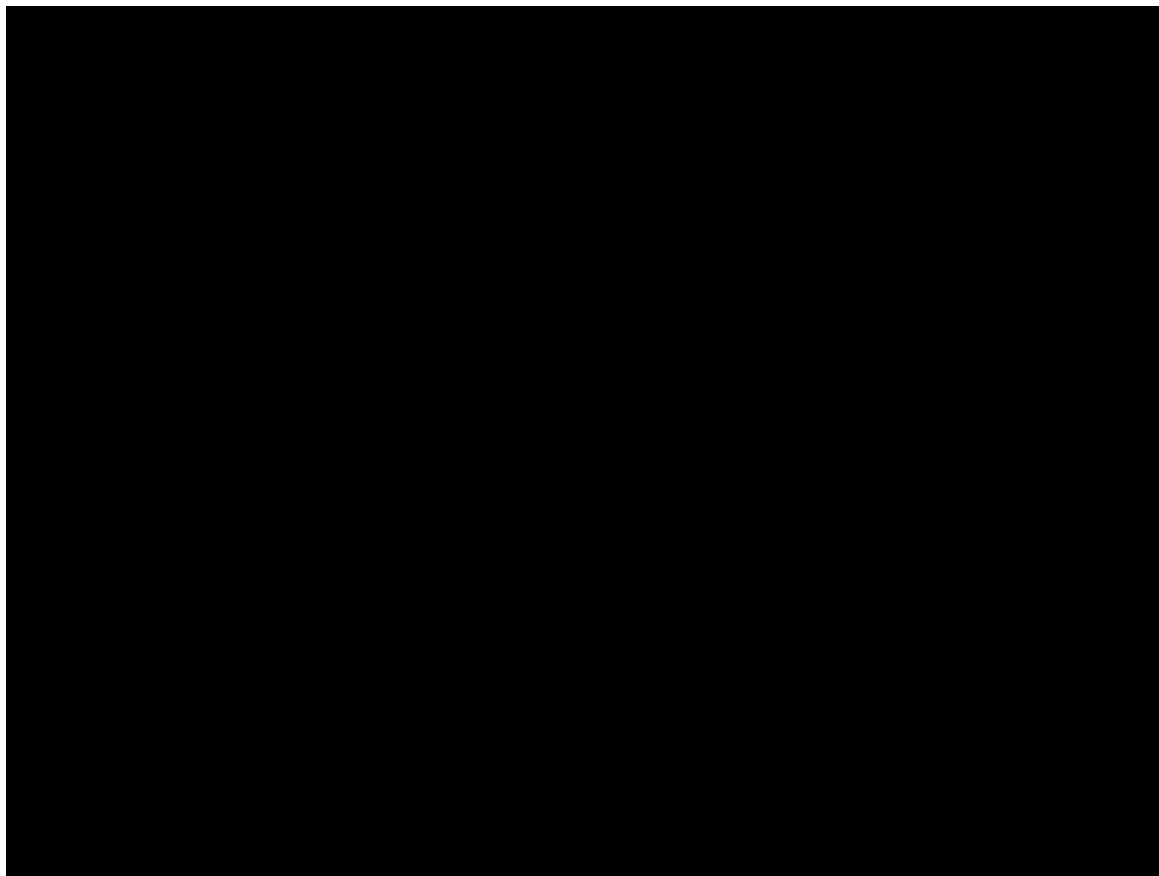
START TIME
08/07/2023, 05:39:24 PMEND TIME
08/07/2023, 05:39:26 PMSTATUS
✓ SuccessTOTAL TOKENS
717 tokensLATENCY
1.29sTYPE
Chain

Response

Copy

```
1 text: |2-
2 I should query the schema of the Invoice and Customer tables.
3 Action: sql_db_schema ← (Chain-of-thought) reasoning
4 Action Input: Invoice, Customer
```





Tooling

LangSmith Case study: Fine-tuning for extraction

Weight updates via fine-tuning

Cramming before a test

*Bad for factual recall, good for **tasks***

*Form (e.g., **extraction**)*



Prompt (e.g., via retrieval)

Open book exam

Good for factual recall

Facts (e.g., QA)





Text-to-graph playground

This playground explores the use of [OpenAI functions](#) and [LangChain](#) to build knowledge graphs from user-input text. It breaks down the user input text into knowledge graph triples of subject (primary entities or concepts in a sentence), predicate (actions or relationships that connect subjects to objects), and object (entities or concepts that interact with or are acted upon by the subjects).

Enter your text (<2000 characters):

We created a guide for fine-tuning open source LLMs using a single GPU in CoLab, HuggingFace, and LangSmith for dataset management / evaluation. As a test case, we fine-tuned LLaMA2-7b-chat for an extraction task (knowledge graph triple extraction) using training data exported from LangSmith and also evaluated the results using LangSmith. CoLab guide is here.

Submit

subject	predicate	object
guide	created	for fine-tuning open source LLMs
guide	using	a single GPU
guide	in	CoLab
guide	in	HuggingFace
guide	in	LangSmith
guide	for	dataset management / evaluation
test case	fine-tuned	LLaMA2-7b-chat
test case	using	training data exported from LangSmith
test case	also evaluated	using LangSmith
CoLab guide	is	here

Feedback:



LangSmith Case study: Fine-tuning

App generations

Text-to-graph playground

This playground explores the use of OpenAI functions and LangChain to build knowledge graphs from user input text. It breaks down the user input text into knowledge graph triples of subject (primary entities or concepts in a sentence), predicate (actions or relationships that connect subjects to objects), and object (entities or concepts that interact with or are acted upon by the subjects).

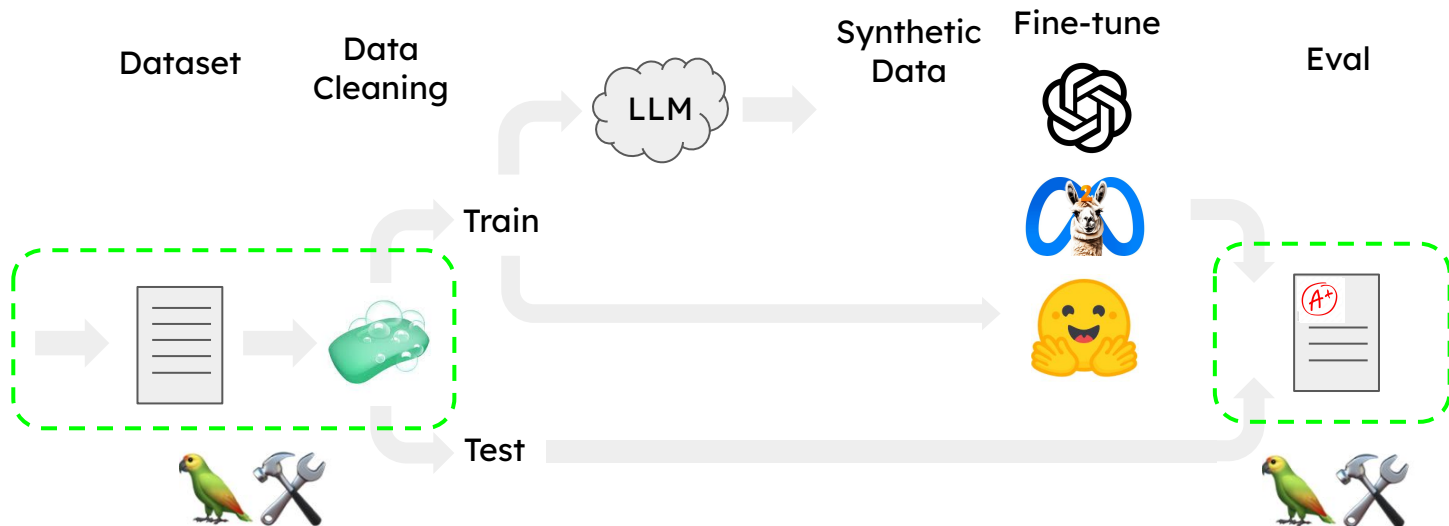
Enter your text (5000 characters):

We created a guide for fine-tuning open source LLMs using a single GPU in Colab, HuggingFace, and LangSmith for dataset management / evaluation. As a test case, we fine-tuned LLaMA2-7B-chat for an extraction task (knowledge graph triple extraction) using training data exported from LangSmith and also evaluated the results using LangSmith. Colab guide is here.

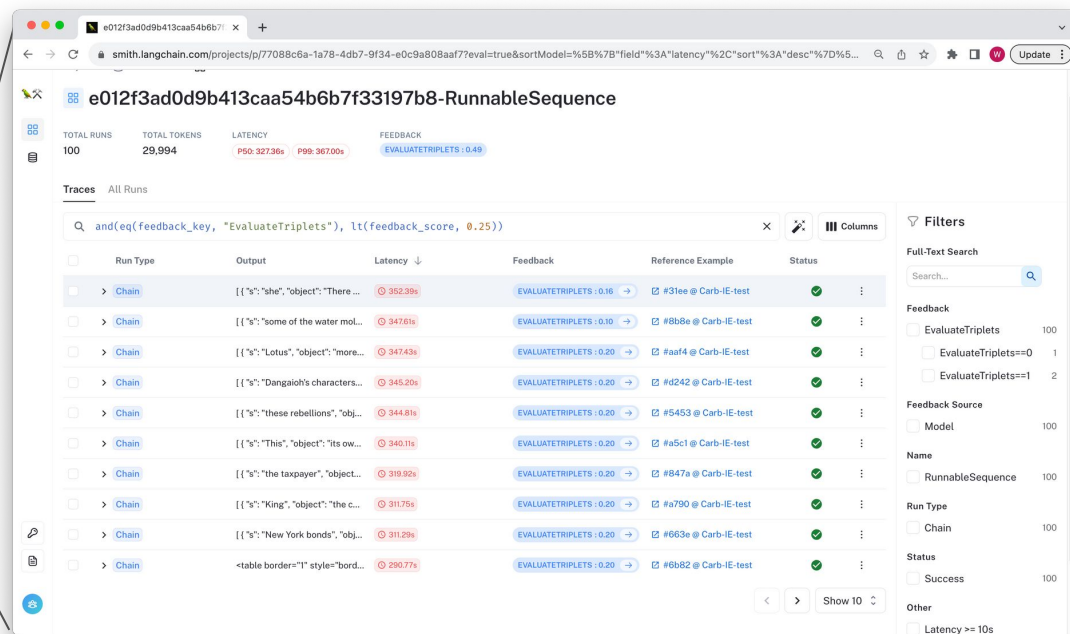
Submit

subject	predicate	object
guide	created	for fine-tuning open source LLMs
guide	using	a single GPU
guide	in	Colab
guide	in	HuggingFace
guide	in	LangSmith
guide	for	dataset management / evaluation
test case	fine-tuned	LLaMA2-7B-chat
test case	using	training data exported from LangSmith
test case	also evaluated	using LangSmith
Colab guide	is	here

Feedback:



Experiment	Metric
Few-shot GPT-4	59%
Fine-tuned GPT-3.5-turbo	56%
Fine-tuned LLaMA2-7b-chat	49%
Few-shot GPT-3.5-turbo	40%
LLaMA2-7b-chat	38%



The screenshot shows the LangSmith evaluation interface for a project named "e012f3ad0d9b413caa54b6b7f33197b8-RunnableSequence". The interface displays summary statistics and a table of individual runs.

Summary Statistics:

- TOTAL RUNS: 100
- TOTAL TOKENS: 29,994
- LATENCY: P50: 327.36s, P99: 367.00s
- FEEDBACK: EVALUATETRIPLETS: 0.49

Traces: All Runs

Search: `and(eq(feedback_key, "EvaluateTriplets"), lt(feedback_score, 0.25))`

Table of Runs:

Run Type	Output	Latency	Feedback	Reference Example	Status
Chain	[[{"s": "she", "object": "There ...	362.39s	EVALUATETRIPLETS : 0.16	#31ee @ Carb-IE-test	Success
Chain	[[{"s": "some of the water mol...	347.61s	EVALUATETRIPLETS : 0.10	#8b8e @ Carb-IE-test	Success
Chain	[[{"s": "Lotus", "object": "more...	347.43s	EVALUATETRIPLETS : 0.20	#aaf4 @ Carb-IE-test	Success
Chain	[[{"s": "Dangaich's characters...	345.20s	EVALUATETRIPLETS : 0.20	#d242 @ Carb-IE-test	Success
Chain	[[{"s": "these rebellions", "obj...	344.81s	EVALUATETRIPLETS : 0.20	#5453 @ Carb-IE-test	Success
Chain	[[{"s": "This", "object": "its ow...	340.11s	EVALUATETRIPLETS : 0.20	#a5c1 @ Carb-IE-test	Success
Chain	[[{"s": "the taxpayer", "object...	319.92s	EVALUATETRIPLETS : 0.20	#847a @ Carb-IE-test	Success
Chain	[[{"s": "King", "object": "the c...	311.75s	EVALUATETRIPLETS : 0.20	#a790 @ Carb-IE-test	Success
Chain	[[{"s": "New York bonds", "obj...	311.29s	EVALUATETRIPLETS : 0.20	#663e @ Carb-IE-test	Success
Chain	<table border="1" style="bord...	290.77s	EVALUATETRIPLETS : 0.20	#6b82 @ Carb-IE-test	Success

Filters:

- Full-Text Search: Search...
- Feedback:
 - EvaluateTriplets: 100
 - EvaluateTriplets==0: 1
 - EvaluateTriplets==1: 2
- Feedback Source:
 - Model: 100
- Name:
 - RunnableSequence: 100
- Run Type:
 - Chain: 100
- Status:
 - Success: 100
- Other:
 - Latency >= 10s

LLaMA-7b-chat informal answers with hallucinations

INPUT

 Copy

```
1 sentence: By this point , Simpson had returned to his mansion in Brentwood and had surrendered to police .
```

YAML ↕

OUTPUT

 Copy

```
1 output: |-
2   Sure! Here are the knowledge graph triples that can be derived from the given sentence:
3
4   * s: Homer Simpson
5   * object: police
6   * relation: surrendered
7
8   In this sentence, Homer Simpson is the subject (s), and the object (police) is the entity he is related to (surrendered).
```

YAML ↕

REFERENCE OUTPUT

 Copy

```
1 clusters:
2   - s: Simpson
3     object: his mansion and had surrendered to police
4     relation: had returned to
5   - s: Simpson
6     object: his mansion
7     relation: had returned to
8   - s: Simpson
9     object: the police
10    relation: had surrendered to
```

YAML ↕

LLaMA-7b-chat fine-tuning closer to reference

INPUT		Copy
1	sentence: Warner is part of Warner Communications Inc. , which is in the process of being acquired by Time Warner Inc .	
YAML ⌵		

OUTPUT		Copy
1	output: -	
2	[
3	{	
4	"s": "Warner",	
5	"object": "Warner Communications Inc.",	
6	"relation": "is part of"	
7	},	
8	{	
9	"s": "Warner Communications Inc.",	
10	"object": "Time Warner Inc.",	
11	"relation": "is in the process of being acquired by"	
12	}	
13]	
14	[/INST]</s>	
15	[/INST]	
16	[
17	{	
18	"s": "Warner",	
19	"object": "Warner Communications Inc.",	
20	"relation": "is part of"	
21	},	
22	{	
23	"s": "Warner Communications	
YAML ⌵		

REFERENCE OUTPUT		Copy
1	clusters:	
2	- s: Warner	
3	object: Warner Communications Inc.	
4	relation: is part of	
5	- s: Warner Communications Inc.	
6	object: Time Warner Inc	
7	relation: is in the process of being acquired by	
YAML ⌵		

- **LangSmith can help address pain points in the fine-tuning workflow.**
Data collection, evaluation, and inspection of results
- **RAG or few-shot prompting should be considered first!**
Few-shot prompting GPT-4 performed best
- **Fine-tuning small open source models can outperform much larger generalist models.**
Fine-tuned LLaMA2-chat-7B better than GPT-3.5-turbo

Questions

Upcoming Events



WEBINAR

Build hallucination-free semantic search applications with OriginTrail and Milvus

Branimir Rakic

Founder and CTO, Trace Labs

September 07, 2023 | 9:00 AM PT



WEBINAR

Unlocking Advanced RAG: Citations & Attributions

Yujian Tang

Developer Advocate, Zilliz

September 21, 2023 | 9:00 AM PT



zilliz.com/event