

# Advancing LLM Capabilities: Rethinking Retrieval-Augmented Generation & AI Automation

---

DINESH RAGHU

IBM RESEARCH

# Outline

---

## 1. Retrieval Augmented Generation

- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- Teaching LLMs to perform Tool-Calling: Challenges and Directions

# Outline

---

## 1. Retrieval Augmented Generation (RAG)

- What is RAG?
- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- Teaching LLMs to perform Tool-Calling: Challenges and Directions

# Parametric LLMs – amazing creative writers!

---



**You**

Narrate the story of rabbit and tortoise in pirate style.

< 5 / 5 >



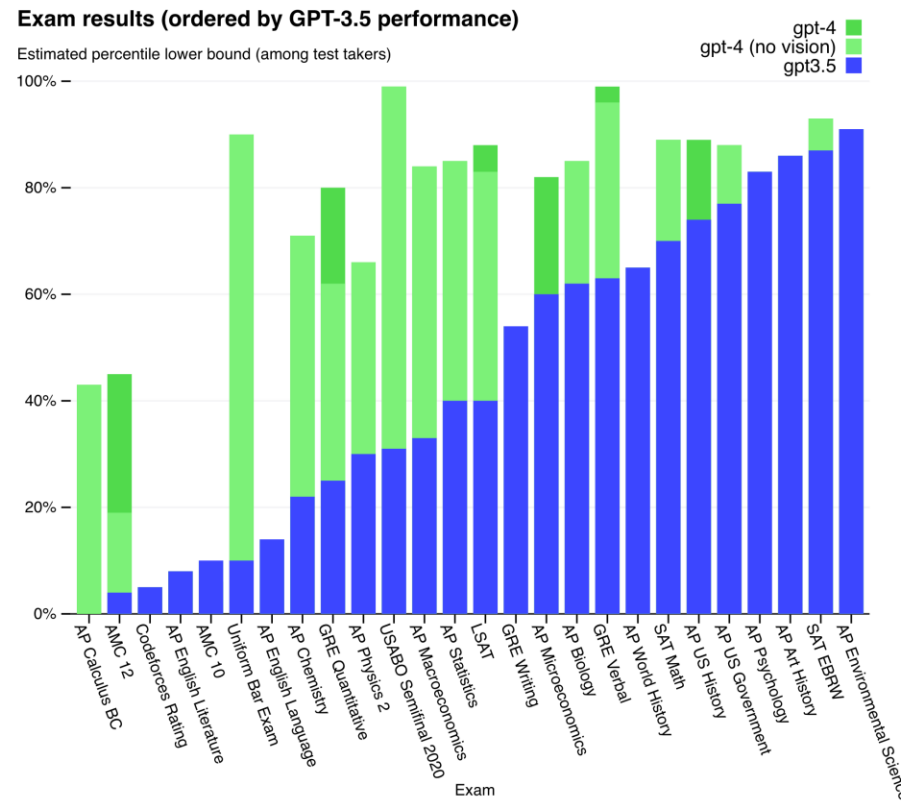
**ChatGPT**

Ahoy there, me hearties! Gather 'round and hear the tale of the great race betwixt the hare and the tortoise, but with a swashbucklin' twist!

Once upon the high seas, there lived a swift and cunning hare named Jackrabbit, known far and wide for his lightning speed and cocky swagger. And in the same waters, there dwelt a slow but steady tortoise by the name of Captain Shellback, renowned for his patience and resilience.

One fine day, as the salty breeze danced through the rigging, Captain Shellback issued a challenge to Jackrabbit, proclaimin', "Arrr, I challenge ye to a race, ye boastful hare! From this here cove to yonder island, we shall see who be the fastest sailor on the seven seas!"

# Parametric LLMs – Stellar performance in exams!



OpenAI – GPT-4 Technical Report  
<https://arxiv.org/abs/2303.08774>

# Hallucinations in Parametric LLMs

---



**You**

what is the use of PMI in NLP?



**ChatGPT**

In natural language processing (NLP), Pointwise Mutual Information (PMI) is often used as a measure of the association between words in a corpus. It can be used to identify words that are highly correlated and occur frequently together, which can be useful for tasks such as language modeling, information retrieval, and machine translation.

For example, in language modeling, PMI can be used to identify the most likely next word in a sentence given the context of the previous words. In information retrieval, PMI can be used to identify the most relevant documents for a given query by ranking documents based on the co-

# Hallucinations in Parametric LLMs

---

Overall, PMI is a useful measure for identifying relationships between words in a corpus and can be applied to a wide range of NLP tasks.



**You**

Can you give me citations for the above examples?



**ChatGPT**

Here are some references for the examples of using PMI in natural language processing (NLP) that I provided:

- Language modeling: "Pointwise Mutual Information (PMI) for Language Modeling" by R. B. Frazer and J. H. Lafferty (<https://www.aclweb.org/anthology/W00-1308.pdf>)
- Information retrieval: "Information Retrieval using PMI-based Word Representations" by T. Mitamura and K. Kise (<https://www.aclweb.org/anthology/C04-1100.pdf>)
- Machine translation: "PMI-based Word Alignment for Statistical Machine Translation" by K. Toutanova and C. D. Manning (<https://www.aclweb.org/anthology/W03-0419.pdf>)

# Hallucinations

Overall, PMI is a useful measure for  
be applied to a wide range of NLP t



You

Can you give me citations for the ab



ChatGPT

Here are some references for the ex  
I provided:

- Language modeling: "Pointwise Mutual Information (PMI) for Language Modeling" by R. B. Frazer and J. H. Lafferty (<https://www.aclweb.org/anthology/W00-1308.pdf>)
- Information retrieval: "Information Retrieval using PMI-based Word Representations" by T. Mitamura and K. Kise (<https://www.aclweb.org/anthology/C04-1100.pdf>)
- Machine translation: "PMI-based Word Alignment for Statistical Machine Translation" by K. Toutanova and C. D. Manning (<https://www.aclweb.org/anthology/W03-0419.pdf>)

About 11,100 results (0.16 sec)

[\[HTML\] PMIVec: a word embedding model guided by point-wise mutual information criterion](#)

[\[HTML\] springer.com](#)

[M Yao](#), [L Zhuang](#), [S Wang](#), [H Li](#) - Multimedia Systems, 2022 - Springer

... the **information** captured by vector similarity between words, this paper proposes a **point-wise mutual information (PMI)**... a masked **language model** during training time, and any **language** ...

☆ Save Cite Cited by 1 Related articles All 4 versions

[A simple language model based on pmi matrix approximations](#)

[\[PDF\] arxiv.org](#)

[O Melamud](#), [I Dagan](#), [J Goldberger](#) - arXiv preprint arXiv:1707.05266, 2017 - arxiv.org

... **language models** by training them to estimate word-context **pointwise mutual information (PMI)**, and then deriving the desired conditional probabilities from **PMI** at test time. Specifically, ...

☆ Save Cite Cited by 17 Related articles All 4 versions

[Unsupervised extractive summarization using pointwise mutual information](#)

[\[PDF\] arxiv.org](#)

[V Padmakumar](#), [H He](#) - arXiv preprint arXiv:2102.06272, 2021 - arxiv.org

... **pointwise mutual information (PMI)** between sentences, which can be easily computed by a pre-trained **language model**... We use the **language model** to compute **PMI**, which then scores ...

☆ Save Cite Cited by 37 Related articles All 6 versions



# Hallucinations

Overall, PMI is a useful method that can be applied to a wide range of tasks.



**You**

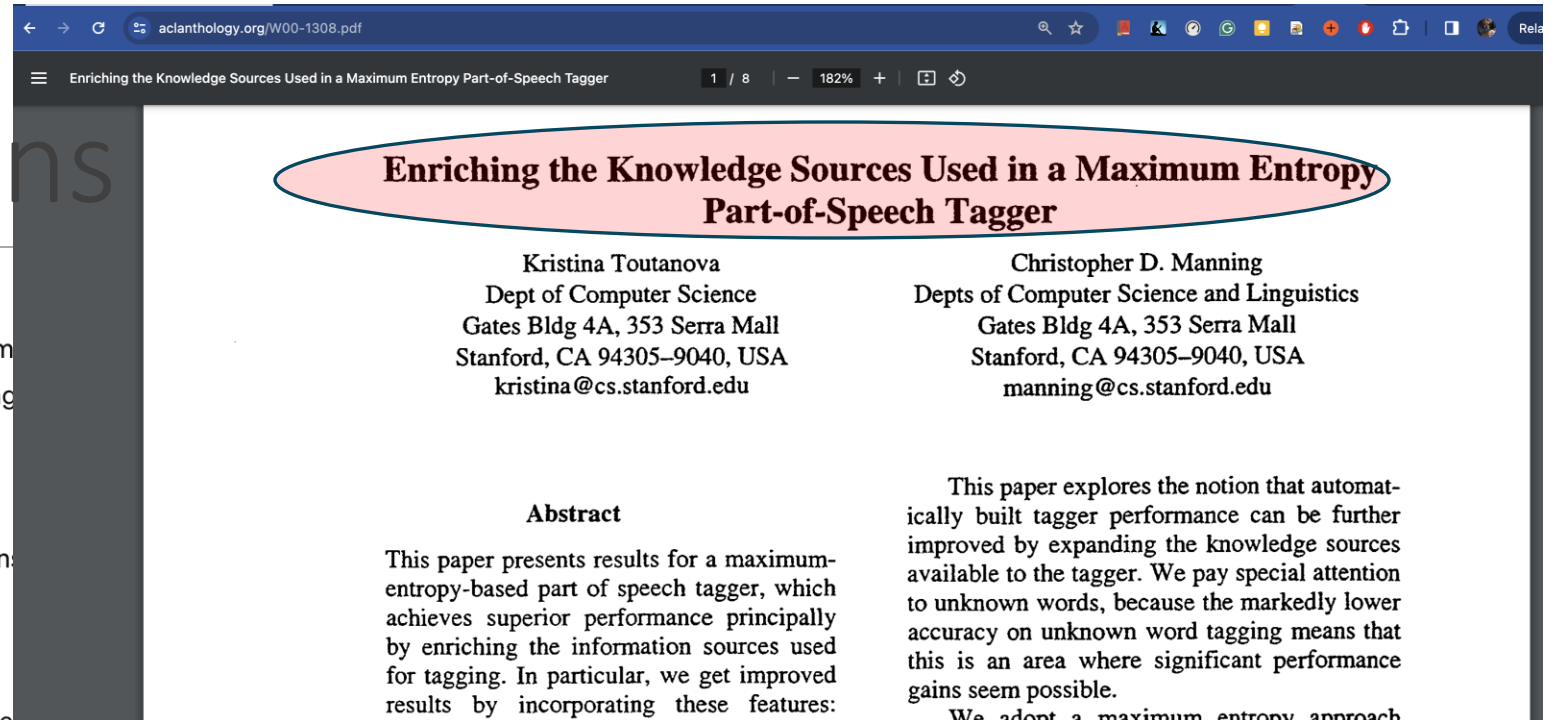
Can you give me citations?



**ChatGPT**

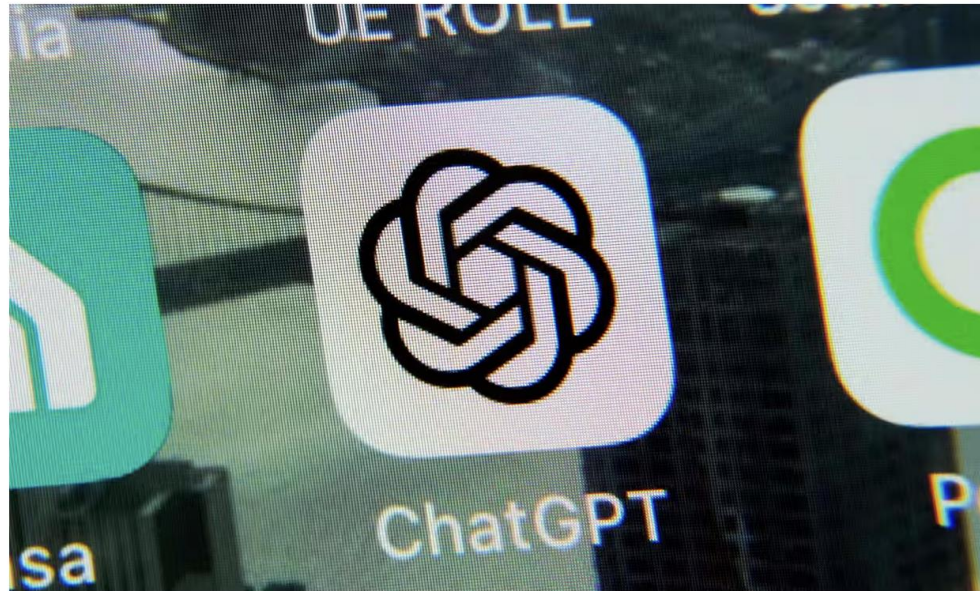
Here are some references for the examples of using PMI in natural language processing (NLP) that I provided:

- Language modeling: "Pointwise Mutual Information (PMI) for Language Modeling" by R. B. Frazer and J. H. Lafferty (<https://www.aclweb.org/anthology/W00-1308.pdf>)
- Information retrieval: "Information Retrieval using PMI-based Word Representations" by T. Mitamura and K. Kise (<https://www.aclweb.org/anthology/C04-1100.pdf>)
- Machine translation: "PMI-based Word Alignment for Statistical Machine Translation" by K. Toutanova and C. D. Manning (<https://www.aclweb.org/anthology/W03-0419.pdf>)



# Two US lawyers fined for submitting fake court citations from ChatGPT

Law firm also penalised after chatbot invented six legal cases that were then used in an aviation injury claim



📷 The judge said one of the fake decisions had 'some traits that are superficially consistent with actual judicial decisions' but other portions contained 'gibberish' and were 'nonsensical'.

Photograph: Richard Drew/AP

A US judge has fined two lawyers and a law firm \$5,000 (£3,935) after fake citations generated by [ChatGPT](#) were submitted in a court filing.

## Air Canada Loses Court Case After Its Chatbot Hallucinated Fake Policies To a Customer

The airline argued that the chatbot itself was liable. The court disagreed.

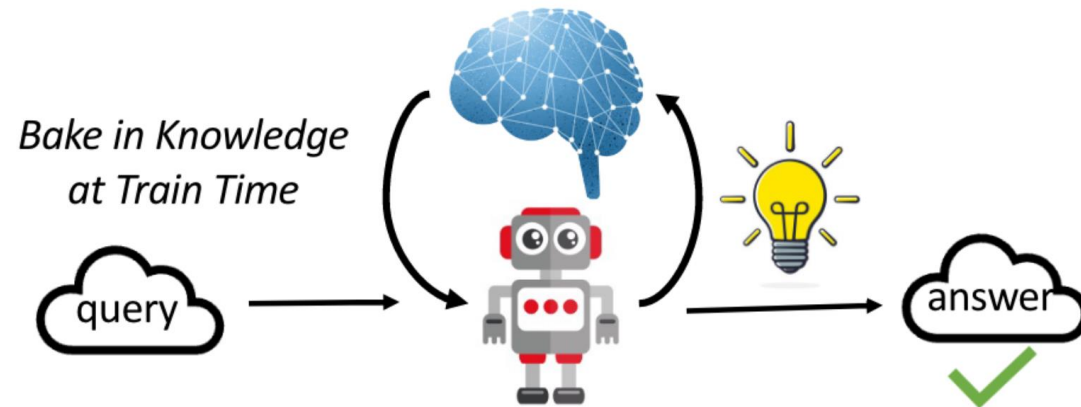
By [Chase Dibeneditto](#) Feb. 18, 2024 f X



# Closed Book vs Open Book Exams

---

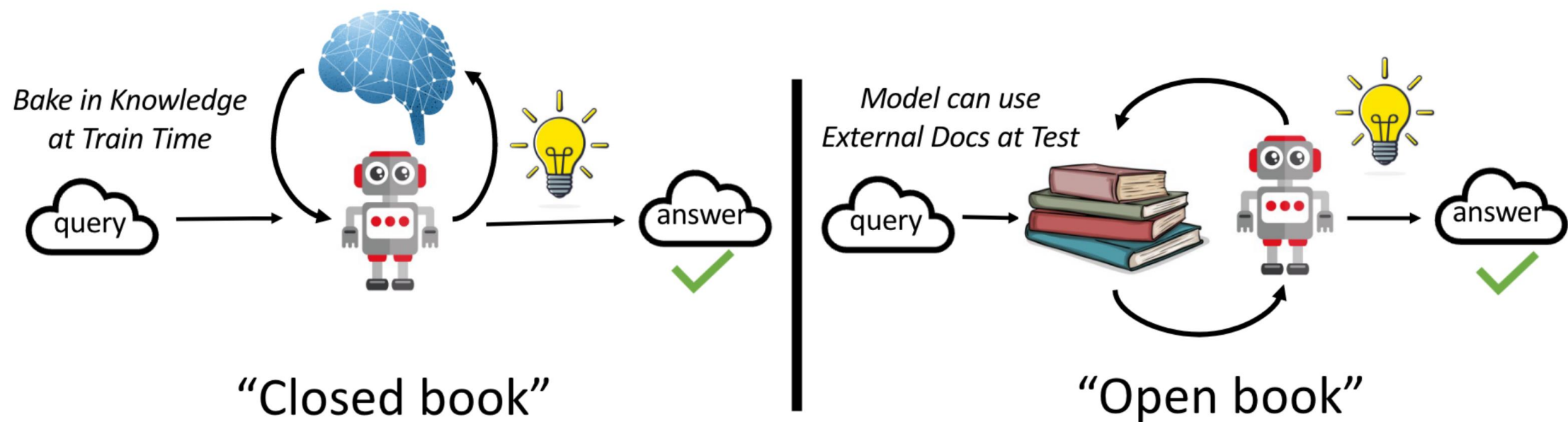
## Parametric LLMs



“Closed book”

# Closed Book vs Open Book Exams

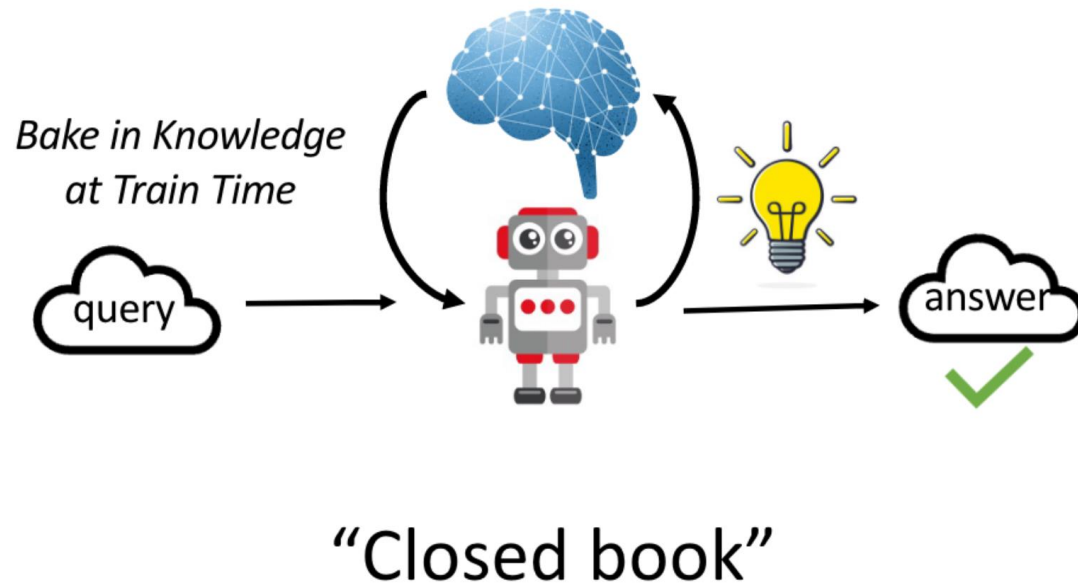
## Parametric LLMs



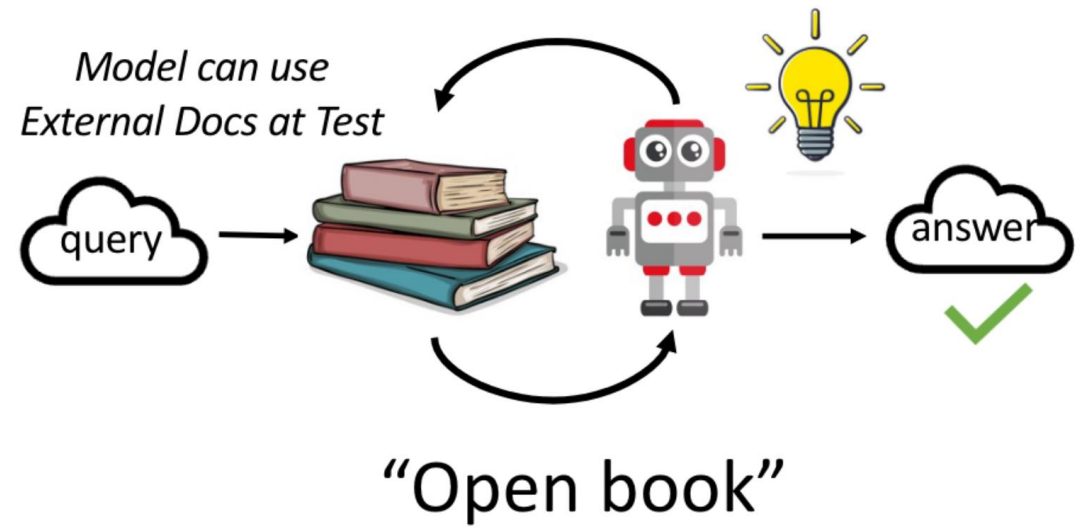


# Closed Book vs Open Book Exams

## Parametric LLMs

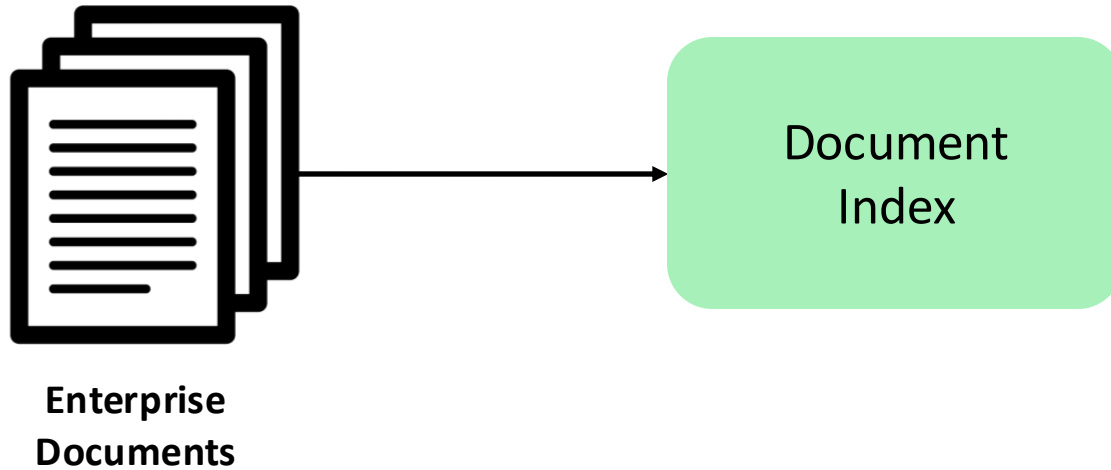


## Retrieval-based LLMs



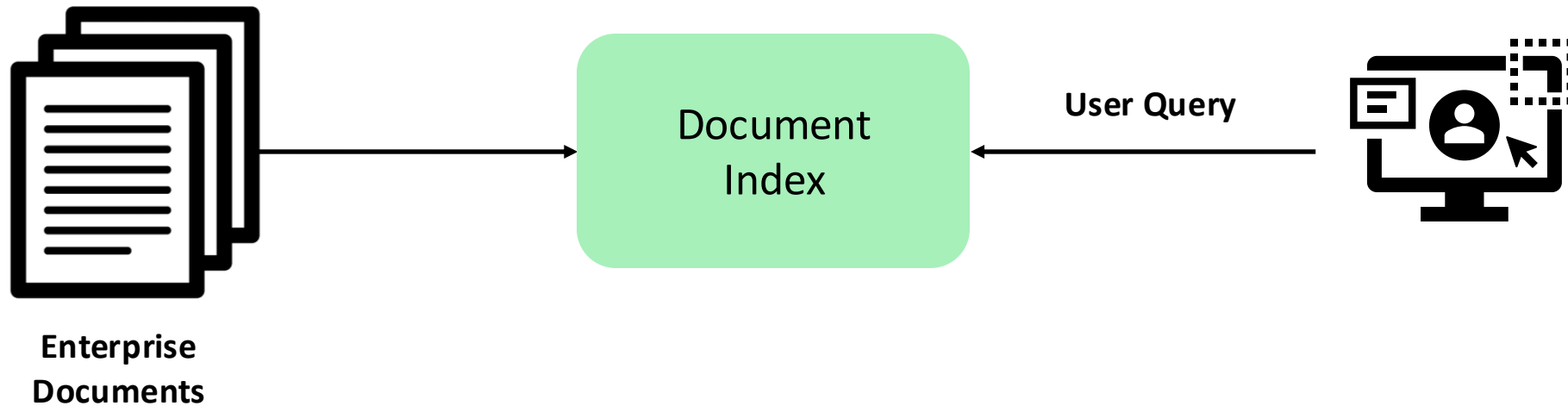
# Retrieval Augmented Generation (RAG)

---



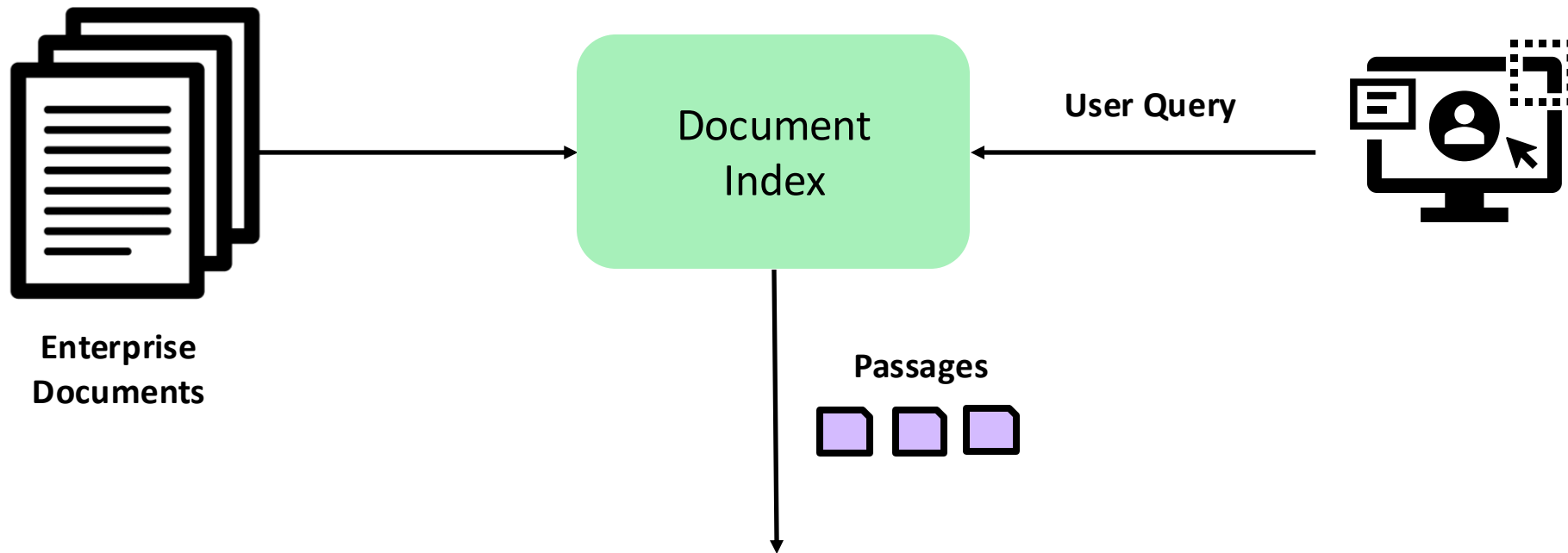
# Retrieval Augmented Generation (RAG)

---



# Retrieval Augmented Generation (RAG)

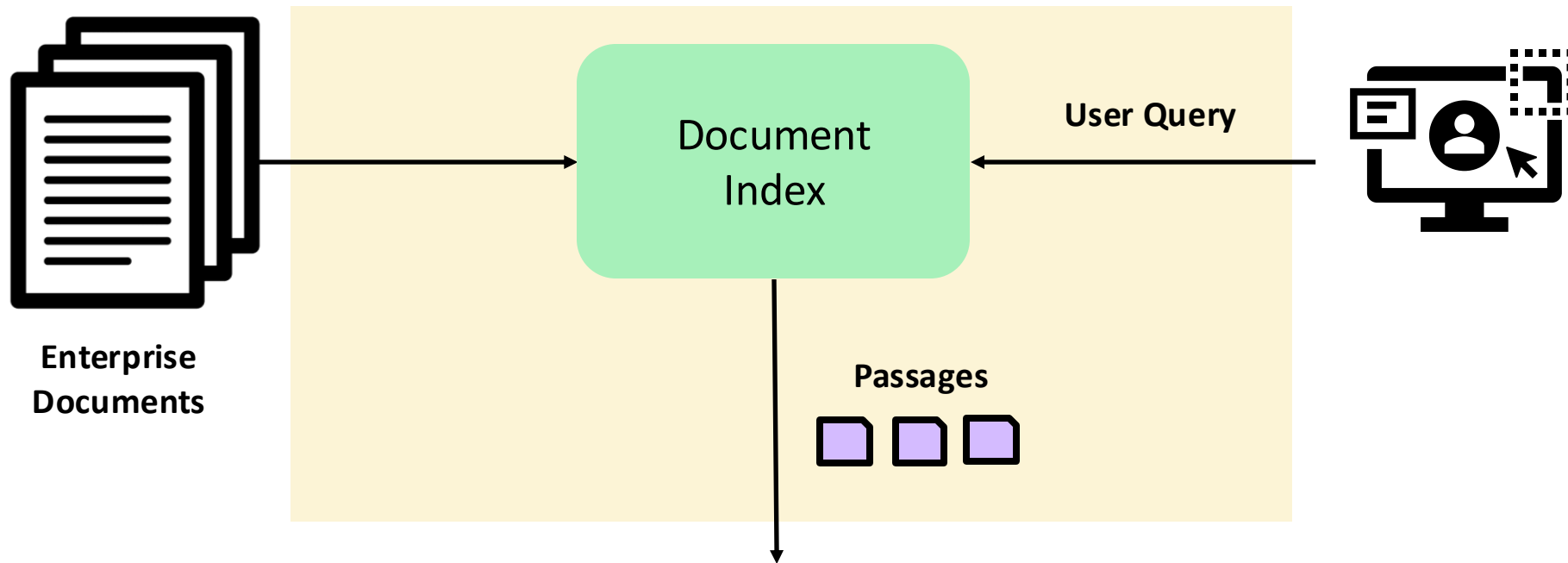
---



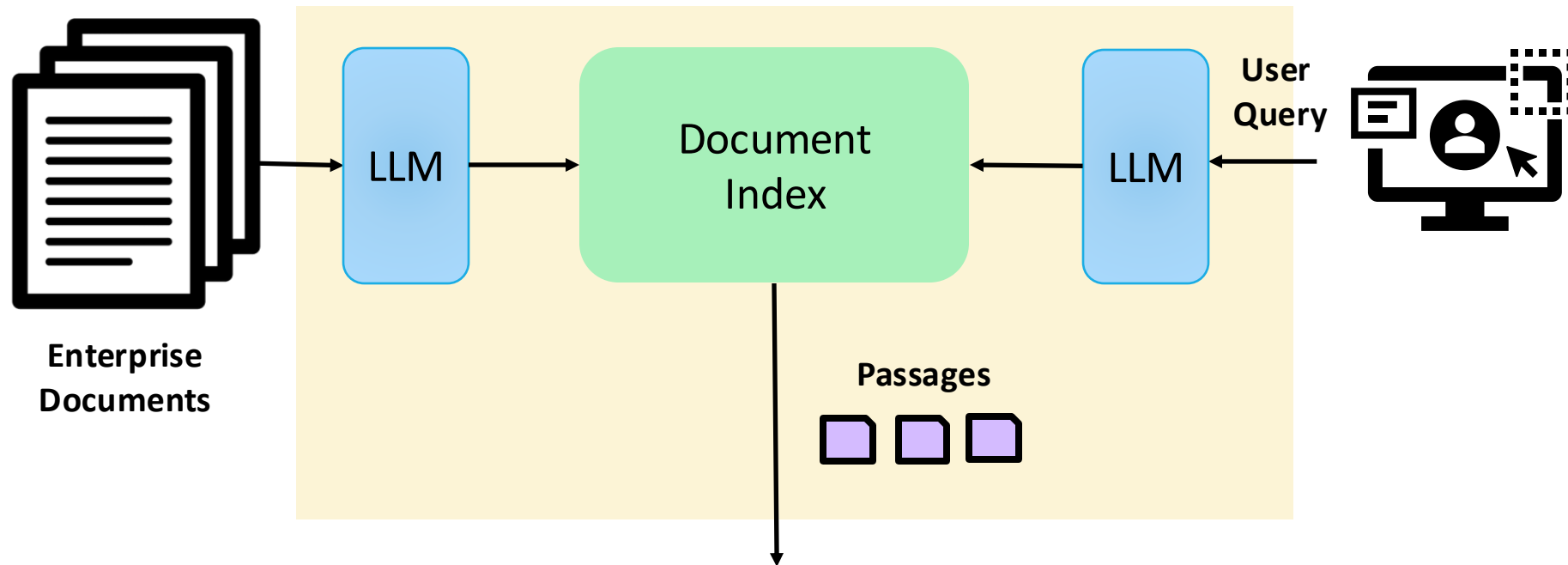


# Retrieval Augmented Generation (RAG)

---

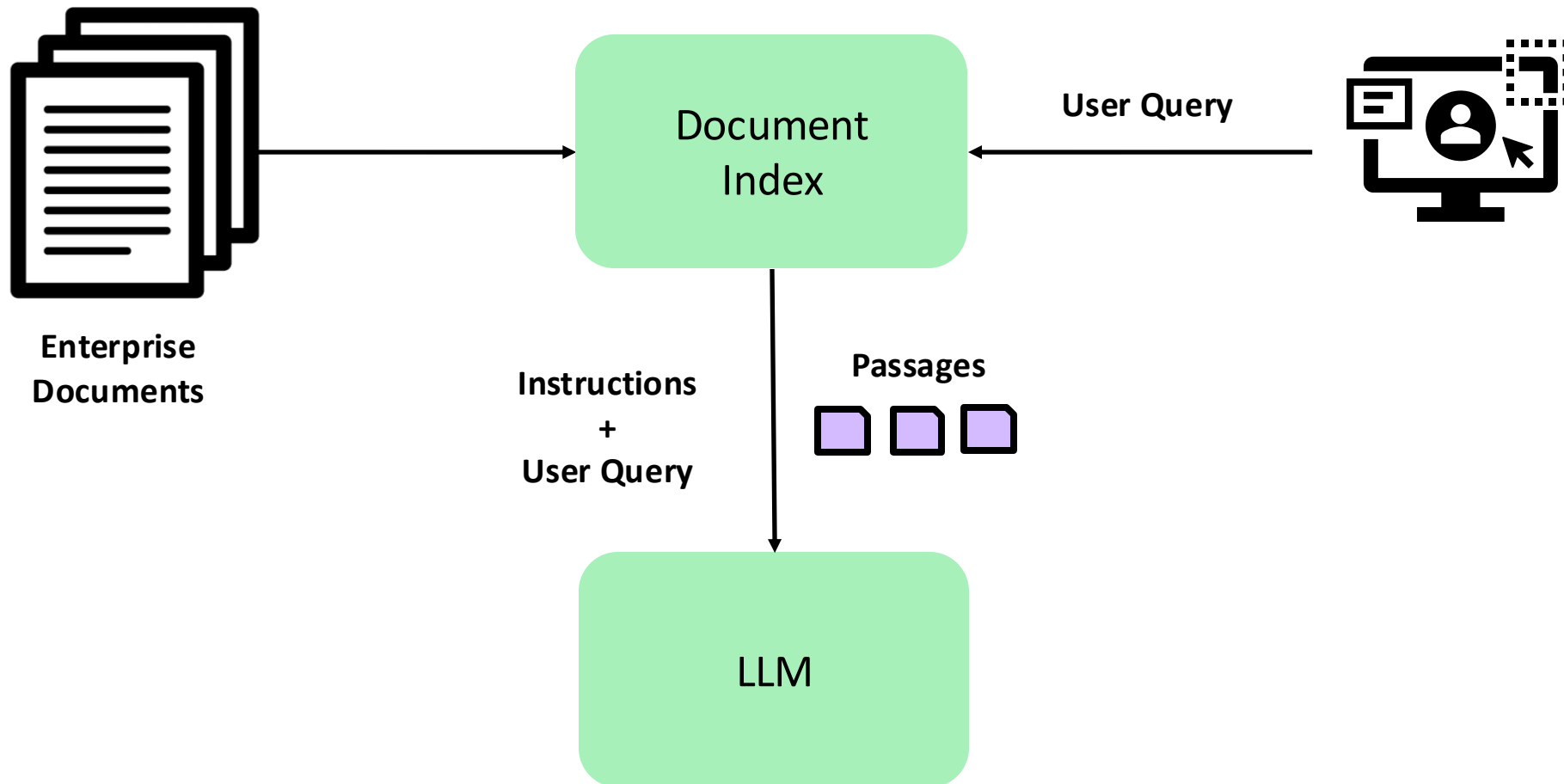


# Retrieval Augmented Generation (RAG)



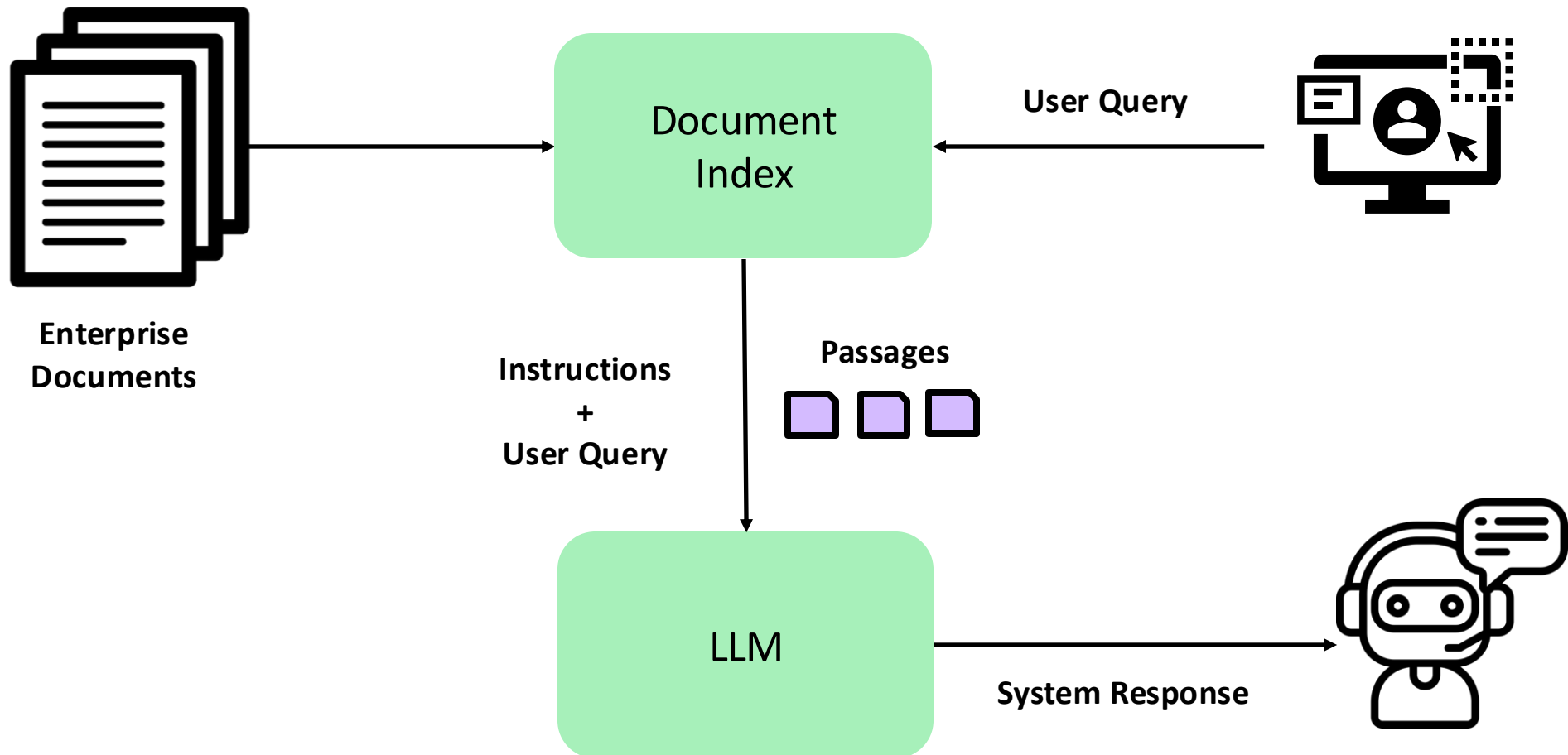
# Retrieval Augmented Generation (RAG)

---



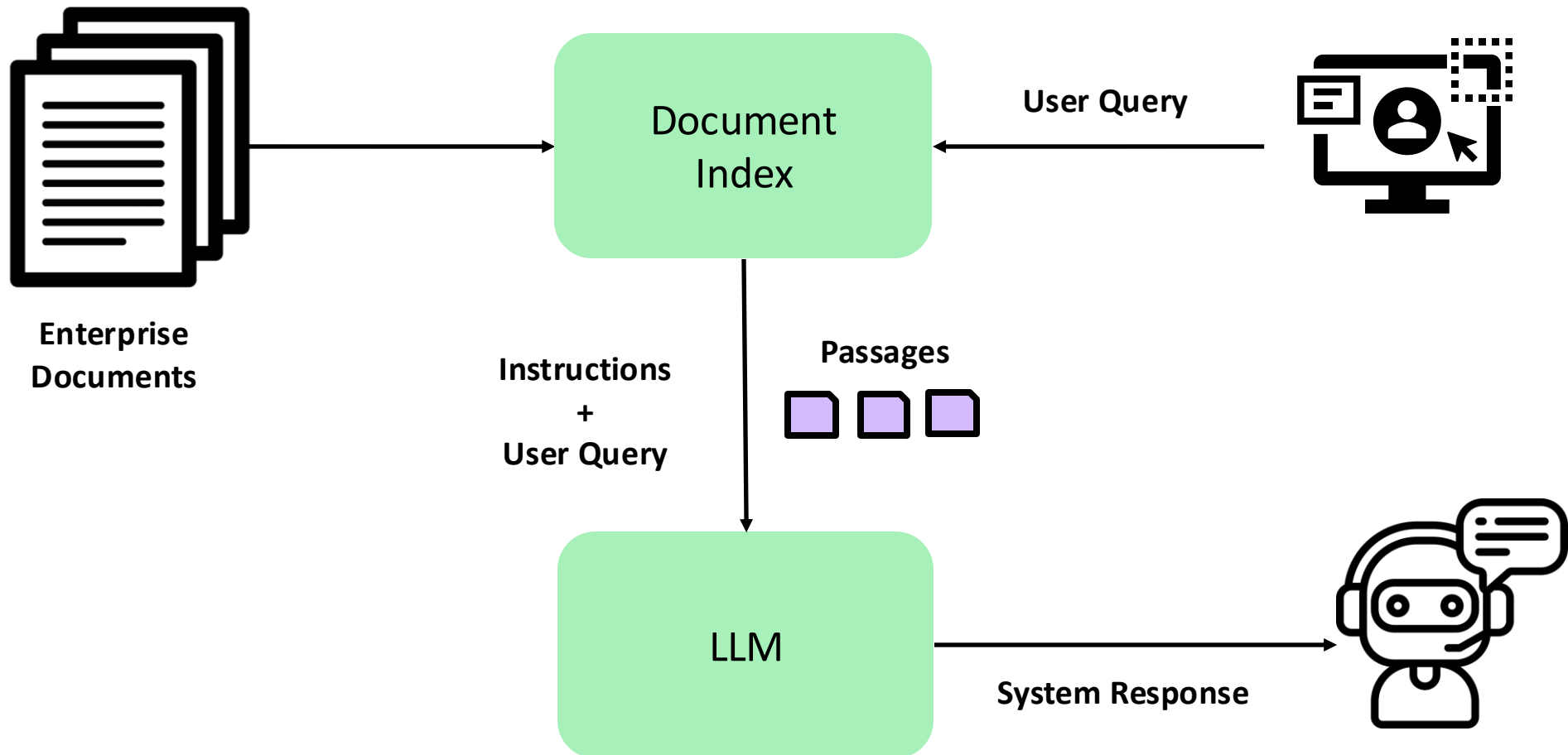
# Retrieval Augmented Generation (RAG)

---



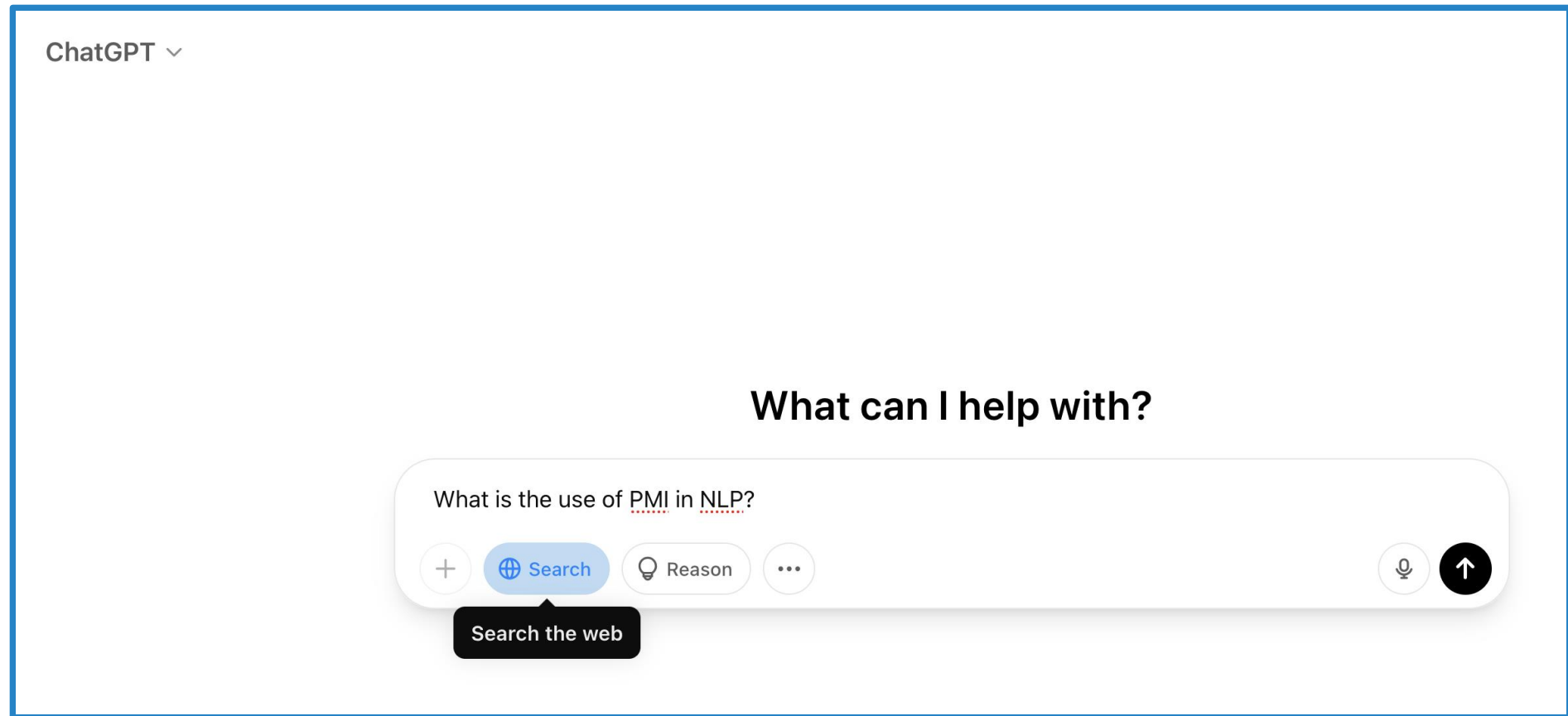
# Retrieval Augmented Generation (RAG)

---



# Retrieval Augmented Generation (RAG)

---



# Retrieval Augmented Generation (RAG)

ChatGPT ▾

What is the use of PMI in NLP?

Share

Pointwise Mutual Information (PMI) is a statistical measure used in Natural Language Processing (NLP) to quantify the association between two words. It assesses how much more (or less) likely two words co-occur than if they were independent. Mathematically, PMI is defined as: [Python in Plain English](#) +2

[towardsmachinelearning.org](#) +3

$$\text{PMI}(x, y) = \log_2 \left[ \frac{P(x, y)}{P(x) \cdot P(y)} \right]$$

[Python in Plain English](#) +1

Here,  $P(x, y)$  is the probability of words  $x$  and  $y$  occurring together, while  $P(x)$  and  $P(y)$  are the individual probabilities of  $x$  and  $y$  occurring independently. [Course Websites](#) +3

**Applications of PMI in NLP:**

- Word Association and Collocation Extraction:** PMI identifies word pairs that occur together more frequently than expected by chance, aiding in discovering collocations like "artificial intelligence" or "machine learning". [ListenData](#)
- Semantic Similarity and Word Embeddings:** By constructing a word-context matrix using PMI values, models can capture semantic relationships between words, enhancing tasks like synonym detection and analogy reasoning.
- Topic Modeling and Information Retrieval:** PMI helps in identifying terms that are distinctive to specific topics or documents, improving the relevance of search results and the coherence of topic models. [Medium](#)
- Sentiment Analysis:** By analyzing the PMI between words and sentiment-laden terms, models can better understand the sentiment conveyed in texts.
- Dialogue Systems and Response Generation:** PMI is utilized to ensure that generated responses in conversational agents are contextually relevant and grounded in the preceding dialogue. [arXiv](#)

**Variants of PMI:**

Ask anything

+

Search

Reason

...

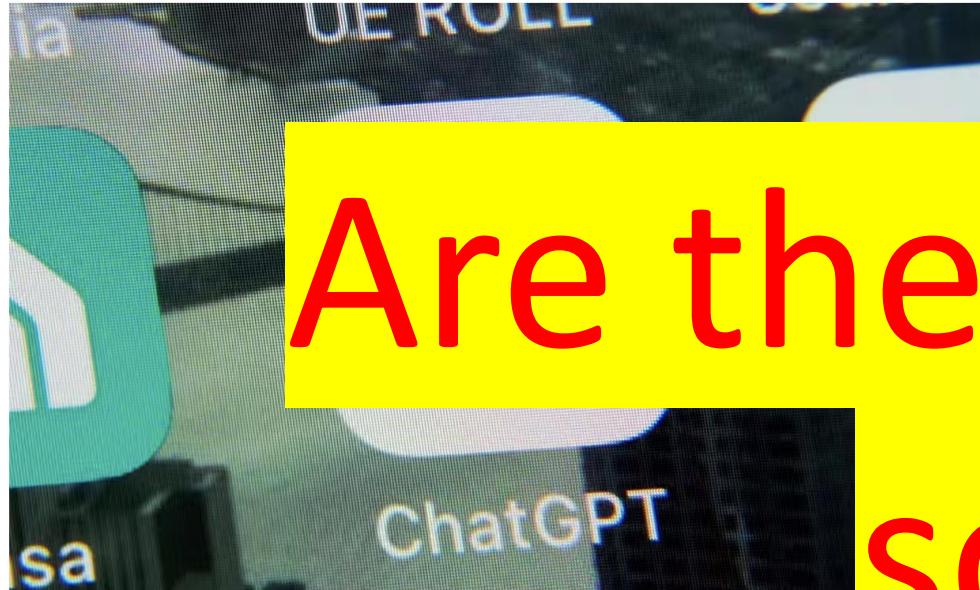
arXiv

Pointwise Mutual Information Based Metric and Decoding Strategy for Faithful Generation in ...

20 May 2023 — A major concern in using deep learning based generative models for document-grounded ...

# Two US lawyers fined for submitting fake court citations from ChatGPT

Law firm also penalised after chatbot invented six legal cases that were then used in an aviation injury claim



The judge said one of the fake decisions had 'some traits that are superficially consistent with actual judicial decisions' but other portions contained 'gibberish' and were 'nonsensical'. Photograph: Richard Drew/AP

A US judge has fined two lawyers and a law firm \$5,000 (£3,935) after fake citations generated by [ChatGPT](#) were submitted in a court filing.

Mashable India

Entertainment Culture Tech Science Mobility Podcast Mashable Deals

Tech

## Air Canada Loses Court Case After Its Chatbot Hallucinated Fake Policies To a Customer

The airline argued that the chatbot itself was liable. The court disagreed.

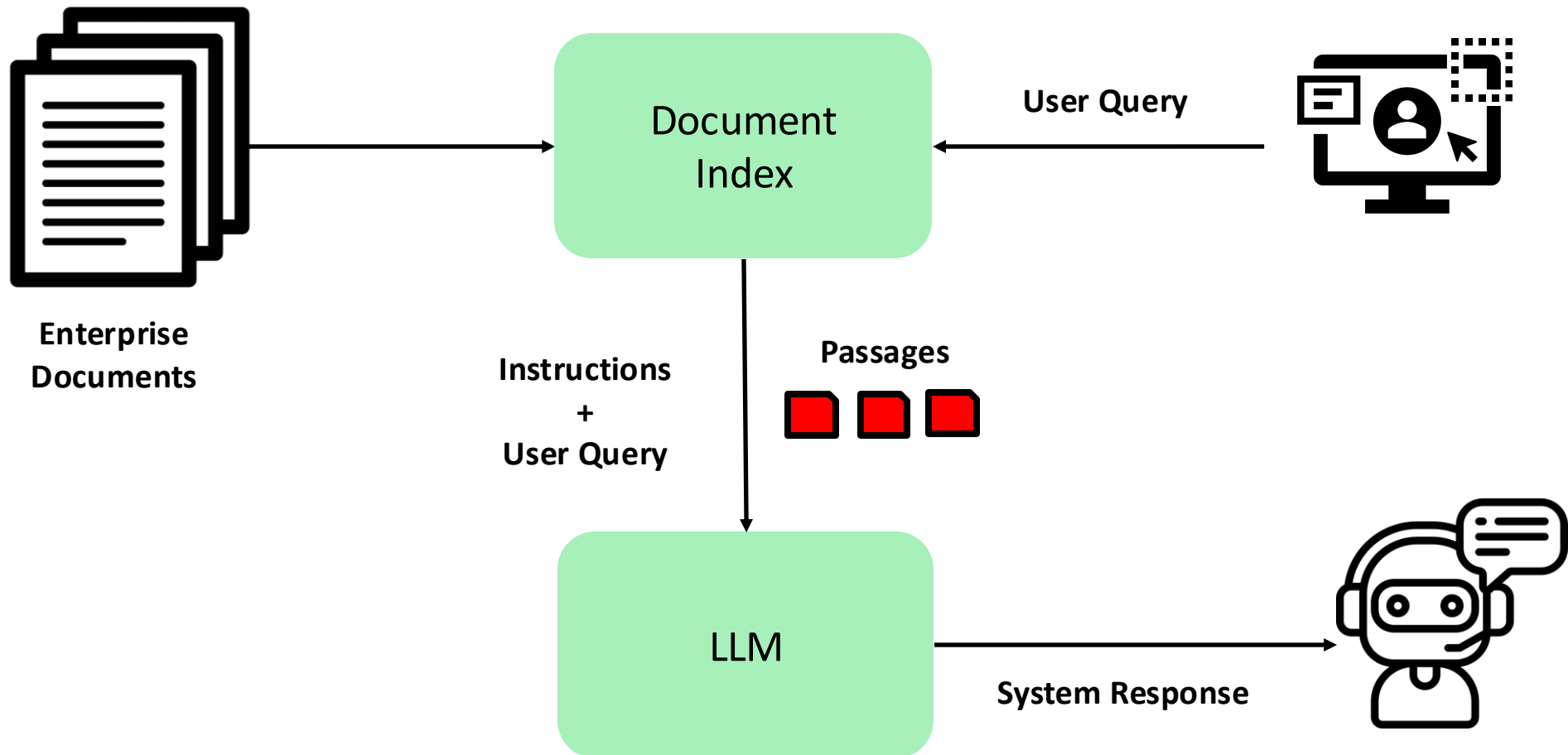
By [Chase Dibeneditto](#) Feb. 18, 2024 f X



Are these problems solved?



# Retriever Failure

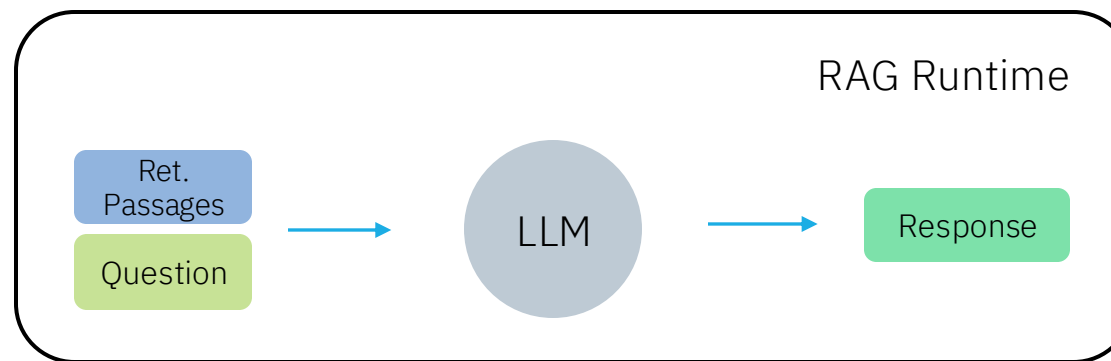


# Retriever Failure

---

In RAG, retriever failure typically results in

1. Incorrect/incomplete responses
2. Hallucinations



## Quick Detour

How can we teach LLM a new task/domain?

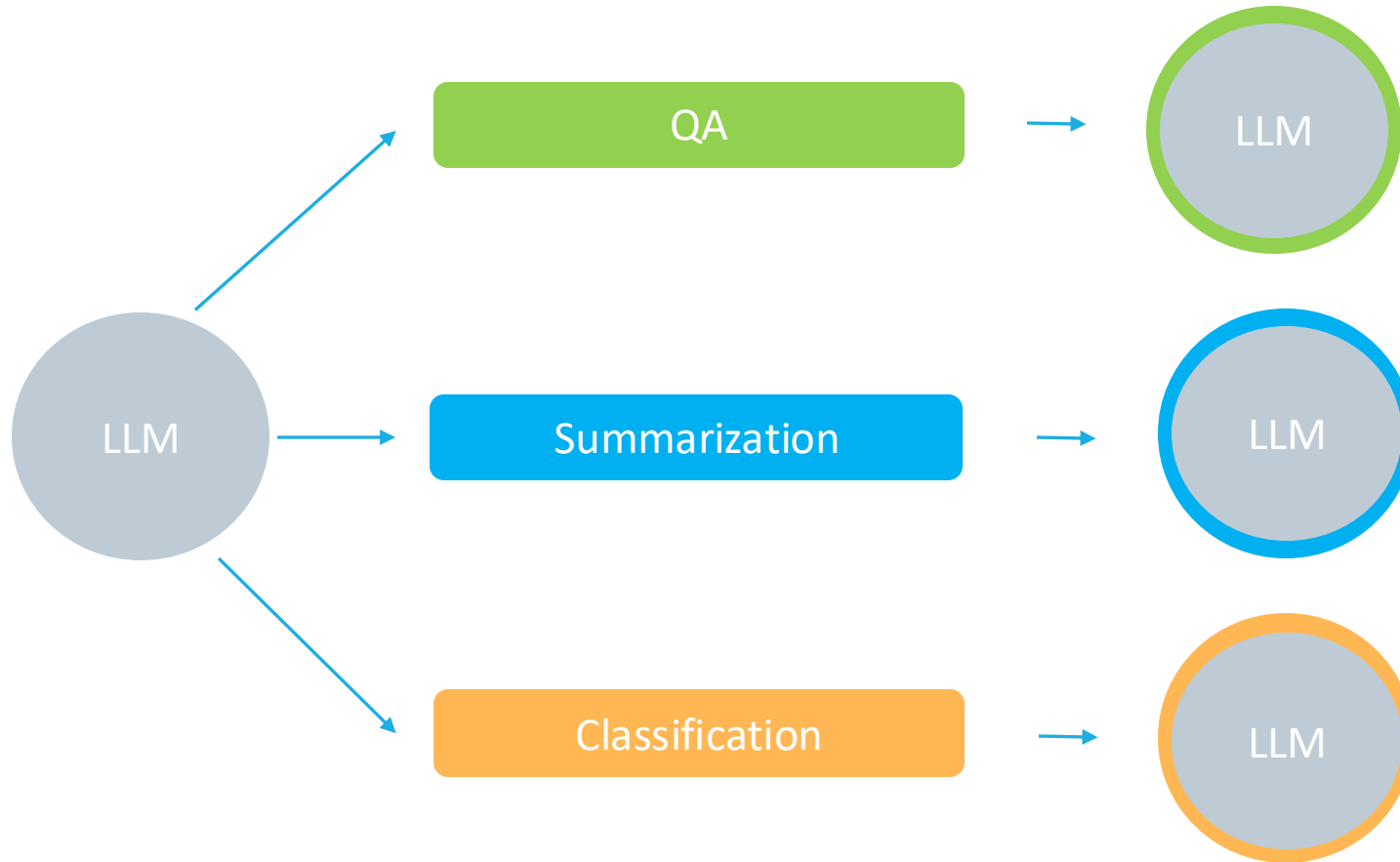
---

# How can we teach LLM a new task/domain?

---

1. Prompt engineering using instructions and in context examples
2. Fine tune the entire model on the task data (aka full finetuning)

# Parameter Efficient Fine Tuning (PEFT)



Popular Solution: PEFT

1. Require less memory than full finetuning
2. Requires less storage than full finetuning

We will use LoRA - a specific method of PEFT for our discussion today.

# Outline

---

## 1. Retrieval Augmented Generation

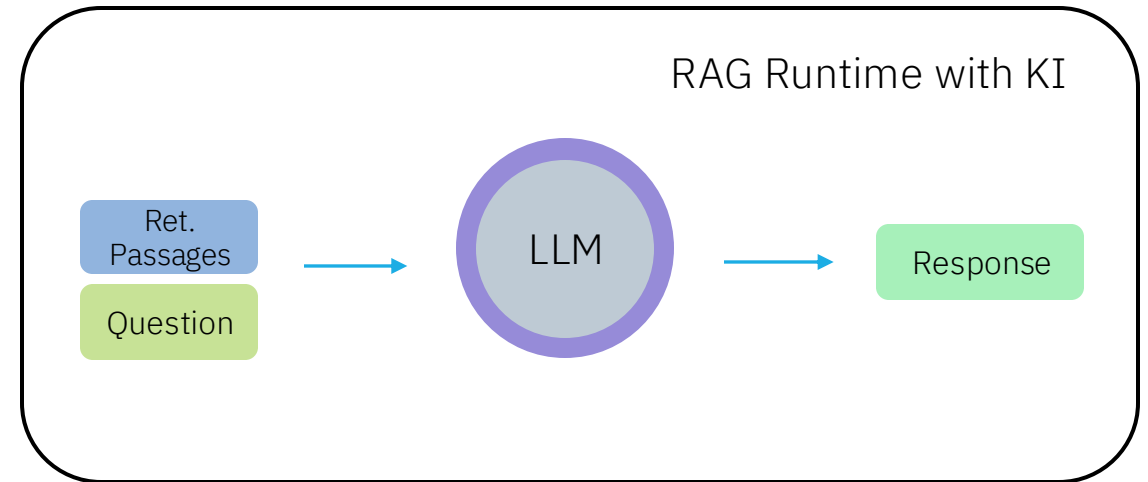
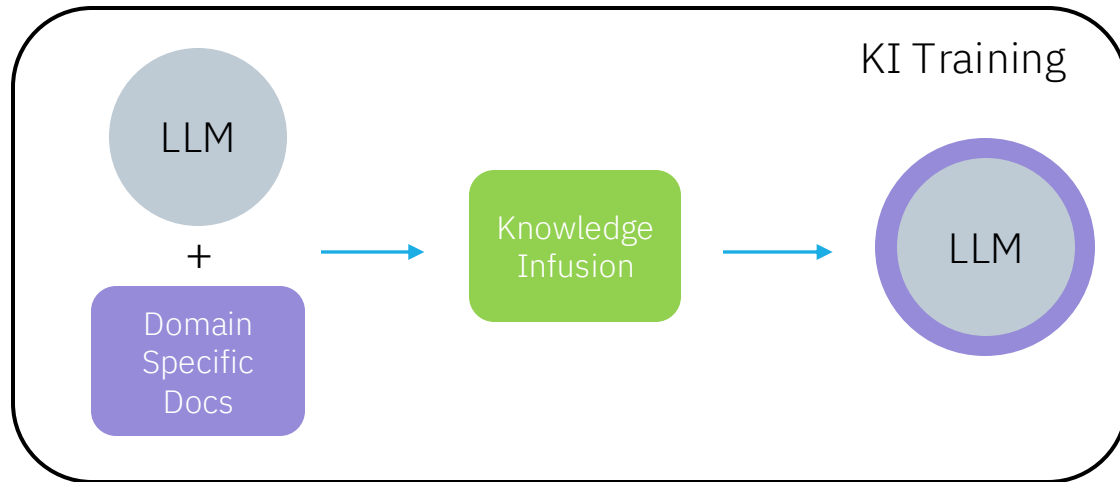
- What is RAG?
- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- Teaching LLMs to perform Tool-Calling: Challenges and Directions

# Knowledge Ingestion in RAG

Ingesting domain knowledge into LLM's parameters may help improve the response correctness when the retriever fails.



# Existing Approaches

---

$q_i$  question

$a_i$  answer



$$q_i + D^* + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



$$q_i + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



# Existing Approaches

$q_i$  question  
 $a_i$  answer



$$q_i + D^* + D_1 + D_2 + \dots + D_k \rightarrow a_i$$












$$q_i + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



\*RAFT: Adapting Language Model to Domain Specific RAG, Zhang et al., COLM 2024

# Issues with RAFT










	RAFT		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$

## Conditional Memorization Bias:

1. The answer  $a_2$  will be memorized, as retrieval failure is simulated for the question  $q_2$
2. But the answers  $a_1$  and  $a_3$  may not be memorized as retrieval success is simulated for the questions  $q_1$  and  $q_3$  respectively.

# Issues with RAFT

---

	RAFT		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$

Canonical Answer Overfitting:

1. Each question in the fine-tuning dataset is associated with only one canonical answer
2. This singular association may lead to learning and replicating **spurious syntactic patterns**










# Proposed Approach












$$q_i + D^* + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



$$q_i + D_1 + D_2 + \dots + D_k \rightarrow a_i$$

	RAFT		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$

	Context Augmentation		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Avoids Conditional Memorization Bias			










# Proposed Approach





















$$q_i + D^* + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



$$q_i + D_1 + D_2 + \dots + D_k \rightarrow a_i$$

	RAFT		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$

	Paraphrase Augmentation		
Epoch 1	 $q_1, a_{11}$	 $q_2, a_{21}$	 $q_3, a_{31}$
Epoch 2	 $q_1, a_{12}$	 $q_2, a_{22}$	 $q_3, a_{32}$
Epoch 3	 $q_1, a_{13}$	 $q_2, a_{23}$	 $q_3, a_{33}$
Avoids Canonical Answer Overfitting			

	Context Augmentation		
Epoch 1	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 2	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Epoch 3	 $q_1, a_1$	 $q_2, a_2$	 $q_3, a_3$
Avoids Conditional Memorization Bias			

# Proposed Approach



$$q_i + D^* + D_1 + D_2 + \dots + D_k \rightarrow a_i$$



$$q_i + D_1 + D_2 + \dots + D_k \rightarrow a_i$$

	RAFT		
Epoch 1	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$
Epoch 2	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$
Epoch 3	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$

	Paraphrase Augmentation		
Epoch 1	$q_1, a_{11}$	$q_2, a_{21}$	$q_3, a_{31}$
Epoch 2	$q_1, a_{12}$	$q_2, a_{22}$	$q_3, a_{32}$
Epoch 3	$q_1, a_{13}$	$q_2, a_{23}$	$q_3, a_{33}$
Avoids Canonical Answer Overfitting			

	Context Augmentation		
Epoch 1	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$
Epoch 2	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$
Epoch 3	$q_1, a_1$	$q_2, a_2$	$q_3, a_3$
Avoids Conditional Memorization Bias			

	Our Approach		
Epoch 1	$q_1, a_{11}$	$q_2, a_{21}$	$q_3, a_{31}$
Epoch 2	$q_1, a_{12}$	$q_2, a_{22}$	$q_3, a_{32}$
Epoch 3	$q_1, a_{13}$	$q_2, a_{23}$	$q_3, a_{33}$
Avoids Conditional Memorization Bias & Canonical Answer Overfitting			

# Results

---

Infused LLMs with contents from 2 books:

- Book 1: [Do More with Less: Automating IBMStorage FlashSystem Tasks with REST APIs, Scripting, and Ansible](#)
- Book 2: [Red Hat OpenShift Container Platform on IBM Z and LinuxONE](#)

Retriever Index Setup:

- ~ 70 books (4765 passages)
- Used LlamaIndex to parse pdfs into *markdown format*
- Used heuristics to extract chapters.
- Each chapter was broken into chunks/passages of 512 tokens
- Added book name, chapter name, and passage number at top of each passage

# Results

---

	Overall	
	Recall	LLM Judge
Base + RAG	67.9	86.2
DSF	70.4	85.4
DSF + RAG	66.6	88.0
RAFT	69.9	87.8
PA-RAG(ours)	<b>77.0</b>	<b>93.8</b>
Base + RAG	69.0	79.2
DSF	69.0	81.9
DSF + RAG	66.5	79.8
RAFT	71.5	85.5
PA-RAG(ours)	<b>75.8</b>	<b>88.0</b>



# Results

	Overall		No overlap		Some overlap	
	Recall	LLM Judge	Recall	LLM Judge	Recall	LLM Judge
Book 1						
Base + RAG	67.9	86.2	56.6	74.6	75.8	94.4
DSF	70.4	85.4	73.0	85.9	68.5	85.1
DSF + RAG	66.6	88.0	67.1	84.5	66.2	90.4
RAFT	69.9	87.8	67.5	85.9	71.5	89.1
PA-RAG(ours)	<b>77.0</b>	<b>93.8</b>	<b>74.0</b>	<b>90.1</b>	<b>79.1</b>	<b>96.4</b>
Book 2						
Base + RAG	69.0	79.2	66.6	75.0	70.6	81.9
DSF	69.0	81.9	69.4	82.2	68.7	81.7
DSF + RAG	66.5	79.8	65.1	79.2	67.5	80.1
RAFT	71.5	85.5	71.3	84.9	71.6	85.9
PA-RAG(ours)	<b>75.8</b>	<b>88.0</b>	<b>75.1</b>	86.7	<b>76.4</b>	<b>88.8</b>

# Issues

---

## 1. Scalability

- a. Difficult to ensure synthetic QA covers the entire corpus
- b. Generating multiple answers for so many questions further adds to the cost

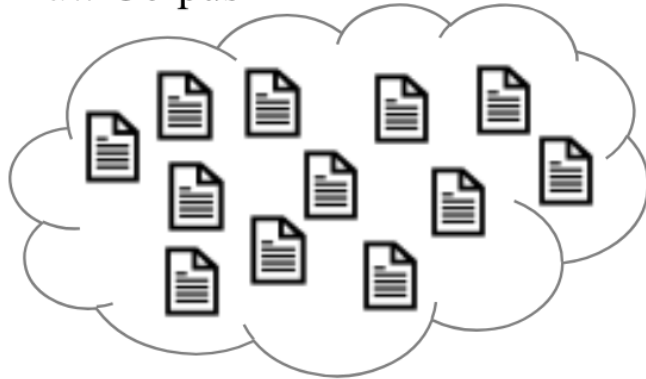
## 2. How to ensure good quality synthetic data?

- a. Vanilla synthetic data generation creates a local myopic view of the corpus

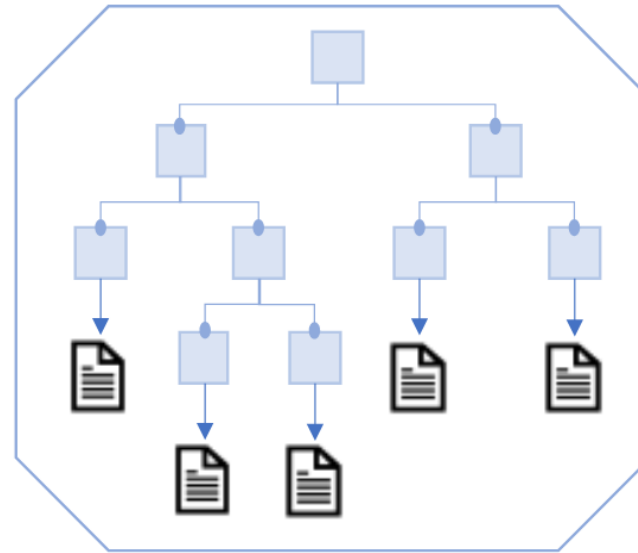
# Issues: local myopic view of the corpus

---

Raw Corpus



Knowledge Structure



# Outline

---

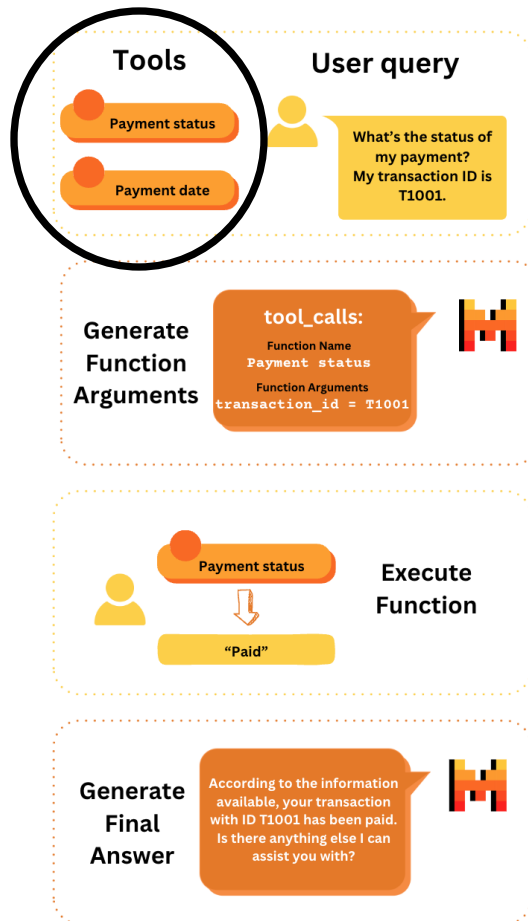
## 1. Retrieval Augmented Generation

- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- What is Tool Calling?
- Teaching LLMs to perform Tool-Calling: Challenges and Directions

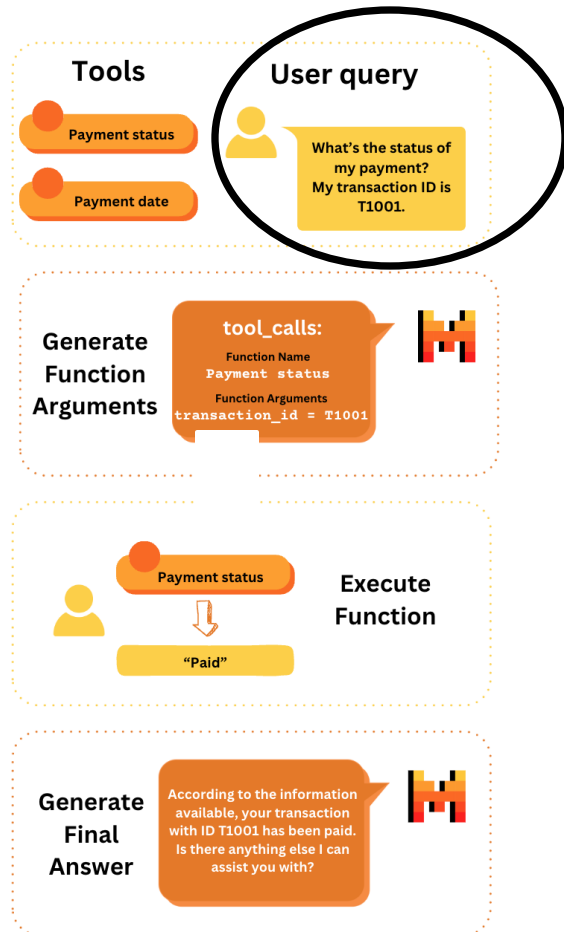
# What is Tool Calling?



**1a. Developer specifies a set of tools and the user enters a query**

```
{
  "type": "function",
  "function": {
    "name": "retrieve_payment_status",
    "description": "Get payment status of a transaction",
    "parameters": {
      "type": "object",
      "properties": {
        "transaction_id": {
          "type": "string",
          "description": "The transaction id.",
        }
      },
      "required": ["transaction_id"],
    },
  },
},
```

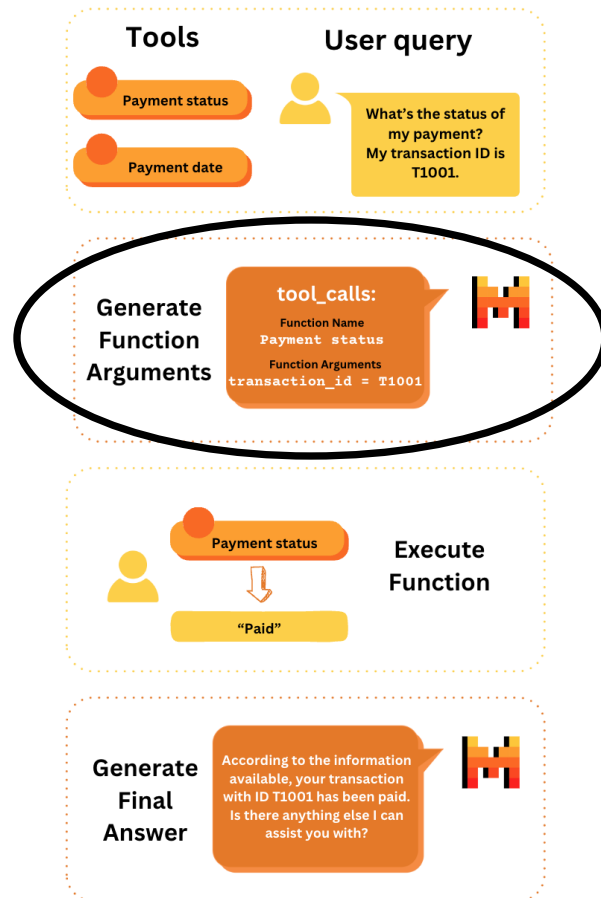
# What is Tool Calling?



## 1b. The user enters a query

What's the status of my payment? My transaction ID is T1001.

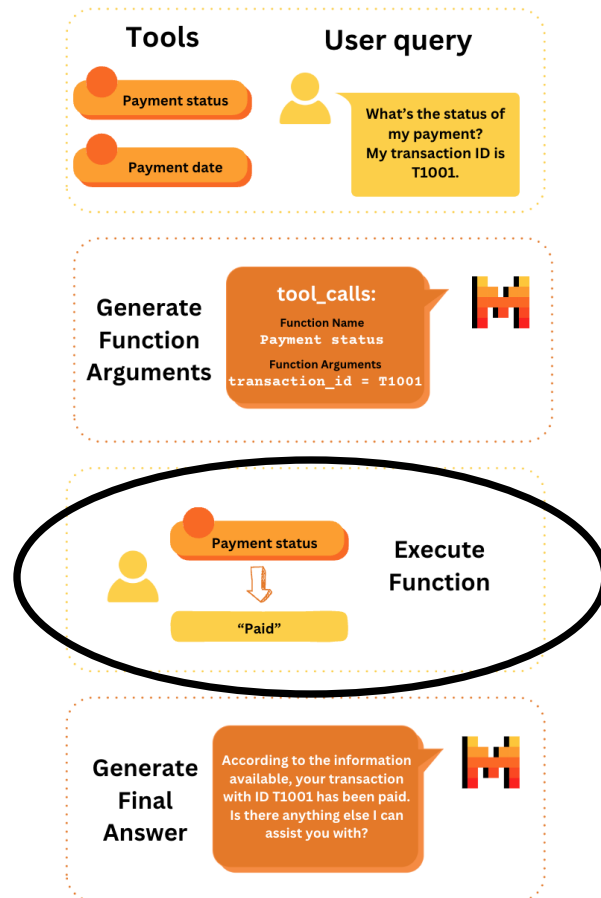
# What is Tool Calling?



## 2. Model identifies the tool and its arguments

```
tool_calls=[ FunctionCall(name='payment_status',
arguments={"transaction_id": "T1001"}) ]
```

# What is Tool Calling?



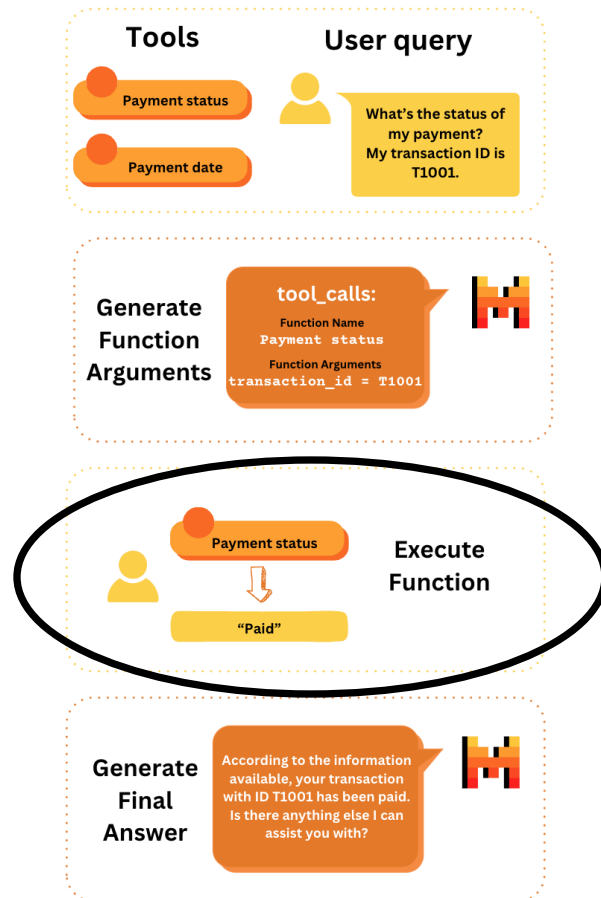
## 3. Runtime executes the tool call to obtain tool results

Function Call Output:

```
{"status": "Paid"}
```



# What is Tool Calling?



## 4. Model uses the tool call results to generate the user response

Function Call Output:

```
{"status": "Paid"}
```

Your transaction with ID T1001 has been paid. Is there anything else I can assist you with?

# Outline

---

## 1. Retrieval Augmented Generation

- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- What is Tool Calling?
- Teaching LLMs to perform Tool-Calling: Challenges and Directions

# Challenges

---

1. Multi Step Tool Calling
2. Support for Long Context
3. Lack of License Friendly Datasets for Training
4. Requires Policy Adherence

# Summary

---

## 1. Retrieval Augmented Generation

- Improve performance by ingesting domain knowledge into LLM parameters

## 2. AI Automation

- Teaching LLMs to perform Tool-Calling: Challenges and Directions

# Conversational AI Team @ IBM Research India

## Knowledge Ingestion for RAG



Yatin  
Nandwani

Anant  
Kumar

Sonam  
Mishra

Meghanadh  
Pulivarthi

Dinesh  
Raghu

Sachindra  
Joshi

## Robust Retrievers for RAG



Vineet  
Kumar

Meghanadh  
Pulivarthi

Sachindra  
Joshi

## LLM Alignment



Gaurav  
Pandey

## Enhancing Tool Calling Capabilities in LLMs



Hima  
Karanam

Pankaj  
Dhoolia

Dheeraj  
Sreedhar

Dinesh  
Khandelwal

G P  
Bhargav

Shajith  
Mohamed

Sumit  
Neelam

Vineet  
Kumar

Dinesh  
Raghu

Sachindra  
Joshi