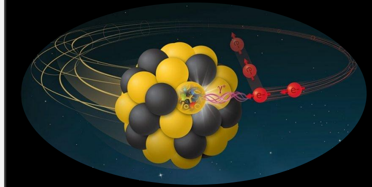


Retrieval Augmented Generation for HEP & NP

Karthik Suresh

College of William and Mary

WILLIAM & MARY
CHARTERED 1693

Image generated by Stable Diffusion XL
From [1]

Outline

- Introduction to Retrieval Augmented Generation (RAG)
- A typical RAG pipeline
- Ingestion methods
- Methods of RAG
- RAG in NEP & HEP
- RAG4EIC effort and roadmap
- How to get involved in RAG4EIC

[GitHub Repo]()

[The application: https://rag4eic.ds.wm.edu](https://rag4eic.ds.wm.edu)

Large Language Models (LLMs)

An LLM is:

- A very advanced autocomplete trained on massive text data
- Takes text in → text out
- Learns patterns, structure, and meaning from human writing
- A compressed snapshot of human knowledge
- Generates, summarizes, reasons – all through text prediction

An LLM is NOT:

- A knowledge base or API – no real-time or factual lookup
- Truly understanding what you ask – just pattern matching
- Aware of meaning or context beyond text correlations
- Reliable and deterministic – same question ≠ same answer

Limitations of LLM

- Not updated to the latest information:
  - Trained on past data only — no real-time updates
  - Requests beyond training cutoff → possible inaccuracies
- Hallucinations:
  - Generates factually wrong or nonsensical answers
  - Output sounds fluent and confident — can mislead users
- Lacks domain-specific accuracy:
  - Performs well on general topics
  - Struggles with specialized, technical, or niche knowledge
- No source citations:
  - Cannot trace where information comes from
  - Hard to verify reliability or reproduce results

arxiv:2005.11401

Retrieval Augmented Generation

The retriever finds relevant documents for a query, and the generator uses those documents to produce the final answer.

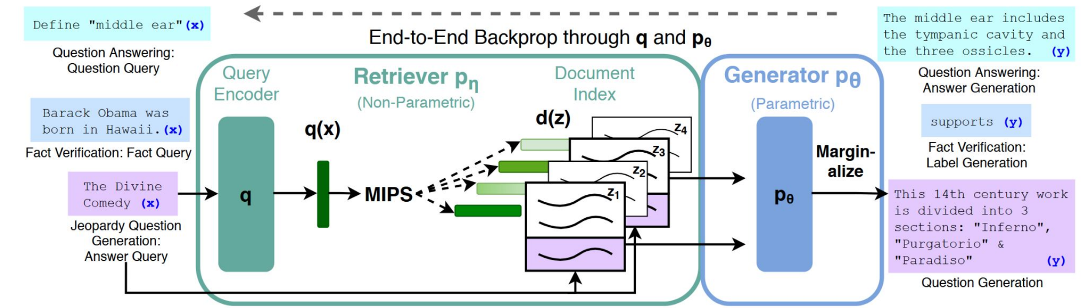
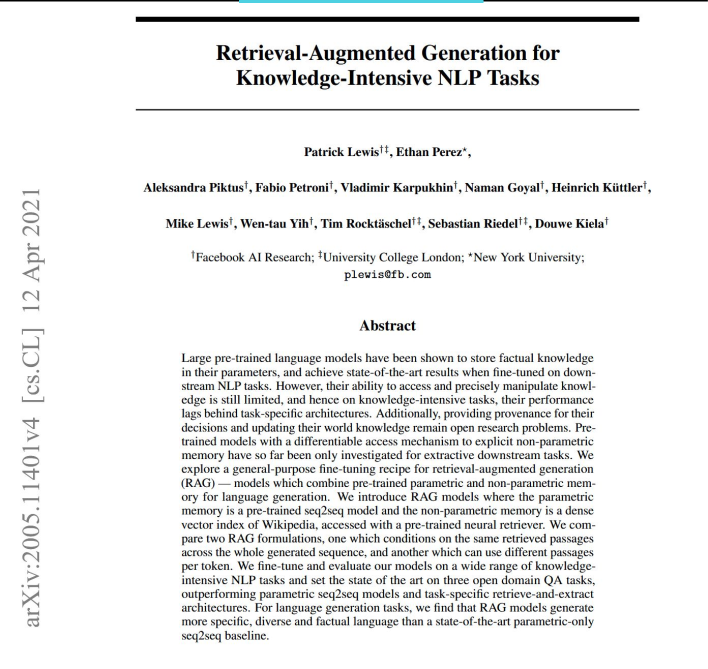


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.



arXiv:2005.11401v4 [cs.CL] 12 Apr 2021

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[⋆],

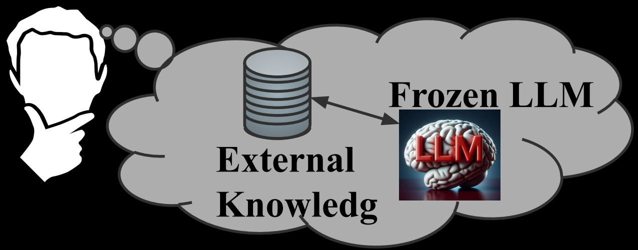
Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel[†‡], Sebastian Riedel[†‡], Douwe Kiela[†]

[†]Facebook AI Research; [‡]University College London; [⋆]New York University;
plewis@fb.com

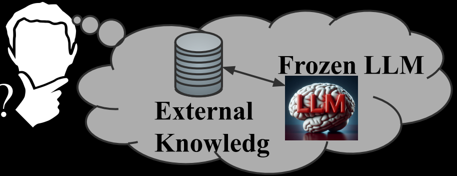
Abstract

Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures. Additionally, providing provenance for their decisions and updating their world knowledge remain open research problems. Pre-trained models with a differentiable access mechanism to explicit non-parametric memory have so far been only investigated for extractive downstream tasks. We explore a general-purpose fine-tuning recipe for retrieval-augmented generation (RAG) — models which combine pre-trained parametric and non-parametric memory for language generation. We introduce RAG models where the parametric memory is a pre-trained seq2seq model and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever. We compare two RAG formulations, one which conditions on the same retrieved passages across the whole generated sequence, and another which can use different passages per token. We fine-tune and evaluate our models on a wide range of knowledge-intensive NLP tasks and set the state of the art on three open domain QA tasks, outperforming parametric seq2seq models and task-specific retrieve-and-extract architectures. For language generation tasks, we find that RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline.



Why RAG based LLM for Large experiments?

Why a summarization tool?



- Ease of retrieving up to date information on topics.
- Extremely useful for new collaborators and especially graduate students.
- Towards building a specialized science for EIC.
- Large Scale Experiments $\propto$ Size of document database.

1400 physicists, 240 institutions and 38 countries….

Why LLM based summary?

- Increased contextual understanding posed by LLM
- Enhanced information retrieval and faster decision making.

The requirement?

- Should be free of hallucinations.
- Should under and unfold all contexts using short prompts
- Provide up to date information without "being" constantly trained

Start with Publicly available documents !!!

Why we need RAG4EIC even for summarizing from public sources?

When asked about a plan to study and become an expert in nuclear structure functions to an LLM with real time RAG capabilities….

That's a fantastic and ambitious goal! Becoming an expert in nuclear structure functions involves understanding how the tiny, dense atomic nucleus is organized and how its components interact. It's a field at the intersection of quantum mechanics and strong nuclear forces, and it helps us understand everything from how stars burn to how nuclear energy is generated.

Let's break down how we can approach this. Here's a learning plan I've put together:

Learning plan:

1. Foundations of Atomic and Nuclear Physics
2. Models of the Nucleus
3. Nuclear Structure Functions
4. Applications and Advanced Topics

By the end of this plan, you'll have a strong grasp of the fundamental concepts, key models, and the "what" and "how" of nuclear structure functions.

Ready to dive in and unravel the mysteries of the atomic nucleus?

This video provides an overview of the cell nucleus, which is a good starting point for understanding the concept of a nucleus in general, even though nuclear structure functions apply to the atomic nucleus. Cell Nucleus Structure & Function

Cell Nucleus Structure & Function - YouTube

Cell Nucleus Structure & Function - YouTube

An Ongoing HEP/NPP effort

AI4EIC-RAG System

Retrieval Augmented Generation System for EIC (RAGS4EIC)

This is a project currently being developed to build a RAG based system for the upcoming Electron Ion Collider

AccGPT - LLMs for CERN!

Vision and Use Case

Florian Rehm, Luke Van Leijenhorst, Verena Kain, Juan Manuel Guijarro, Sofia Vallecorsa

01.07.2025 - IML Machine Learning Working Group: Chatbots at CERN

chATLAS

An AI assistant for the ATLAS collaboration

Daniele Dal Santo[1], Juerg Beringer[2], Joe Egan[3], Ben Elliot[3], Gabriel Facini[3], Daniel Murnane[2,4], Samuel Van Stroud[3], Alex Sopio[5], Jeremy Couthures[6], Joe George[3], Runze Li[7], Cary Randazzo[8]

IML Working Group, 1st July 2025

(1) (2) (3) (4) (5) (6) (7) (8)

HEPilot: Integrating Embeddings, Vector Search, and RAG for Physics Use Case

Mike Sokoloff[1] Conor Henderson[1] Mohamed Elashri[1]

[1]University of Cincinnati
LHCb Collaboration

July 1, 2025

Towards Control Room Automation with a Chatbot for DAQ and Shifter Assistance

Luigi Podda,

with G. Avolio, S. Chapeland, J. Hoya, A. Marzin.

1 July 2025

Retrieval Augmented Generation using LLM pipeline

[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks]

Ingestion

- Load data (semantic) both structured and unstructured from sources
- Split data in small repetitive chunks – text corpus
- Embed these chunks into a vector space using an embedding model
- Store these vectors in a database for retrieval later.

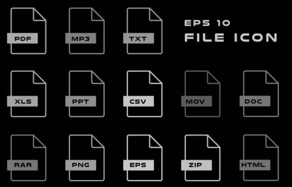
Inference

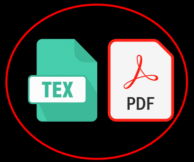
- Given a prompt, compute similarity index. Select the most closest vectors
- Choose a response template. Embed the vectors along with input prompt and feed into LLM
- Evaluate the response
    - Model fine-tuning
    - Build metrics for context tuning
- Multi Modal Output

Ingestion

- The Objective – Summarization tool with "relevant citations" for within EIC.
- Framework built has to be scalable and fairly automated

Data loading





ArXiv publications
Source files and PDFs

Split in Chunks

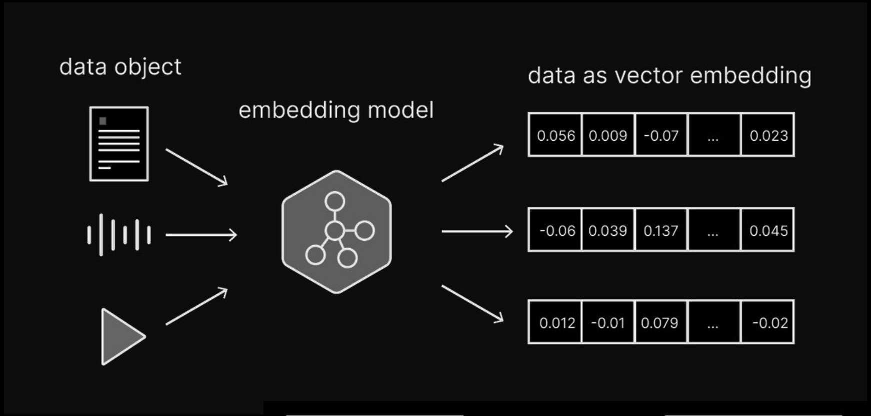
- **PyPDF Reader to read texts from PDF files**
- **Latex splitter – tag based splitting to create chunks. Had to extend and build**
- **Equations selected as an object in Latex splitter**
- **Storing Figure location and table location as metadata while captions are split in chunks – can create multi modal output***

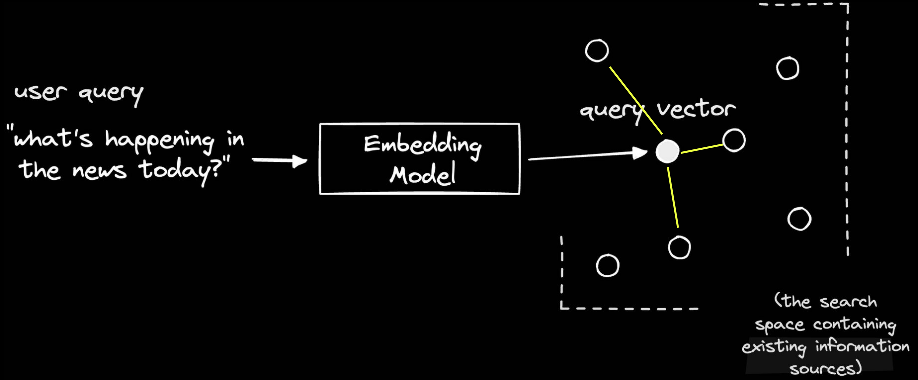
Vector embedding and storage

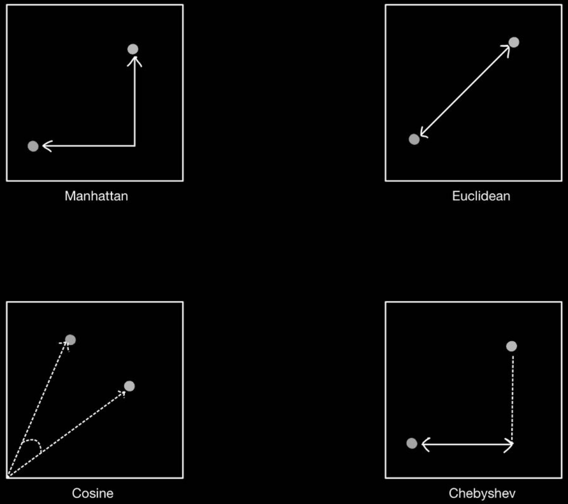
- **Converts any sized input into a fixed size vector**
- **Eg: A 1536 size vector with cosine similarity**
- **Get first 100 similar vectors**
- **LanceDB – Lightweight local DB. Ideal for prototyping.**
- **Scalable cloud: solution PineCone**

The Embedding Model

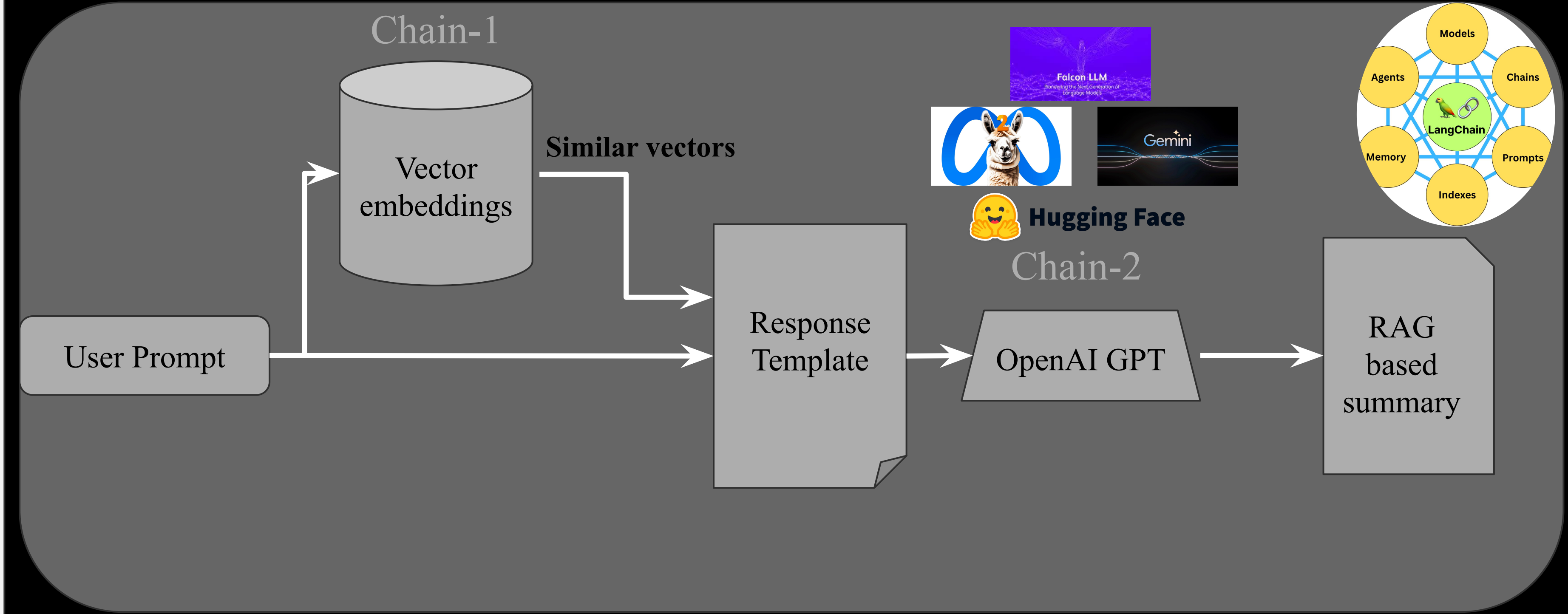
- Embedding models are usually trained with contrastive learning or cosine similarity objectives
- Key function – Preserve semantic similarity



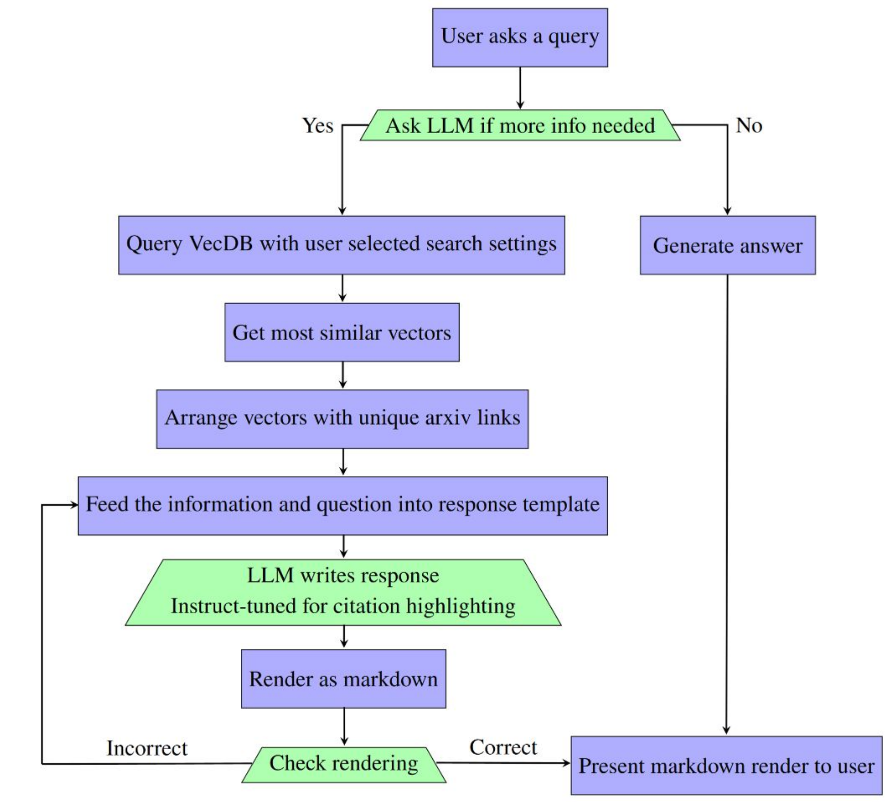




The Generic Workflow of building Traditional RAG

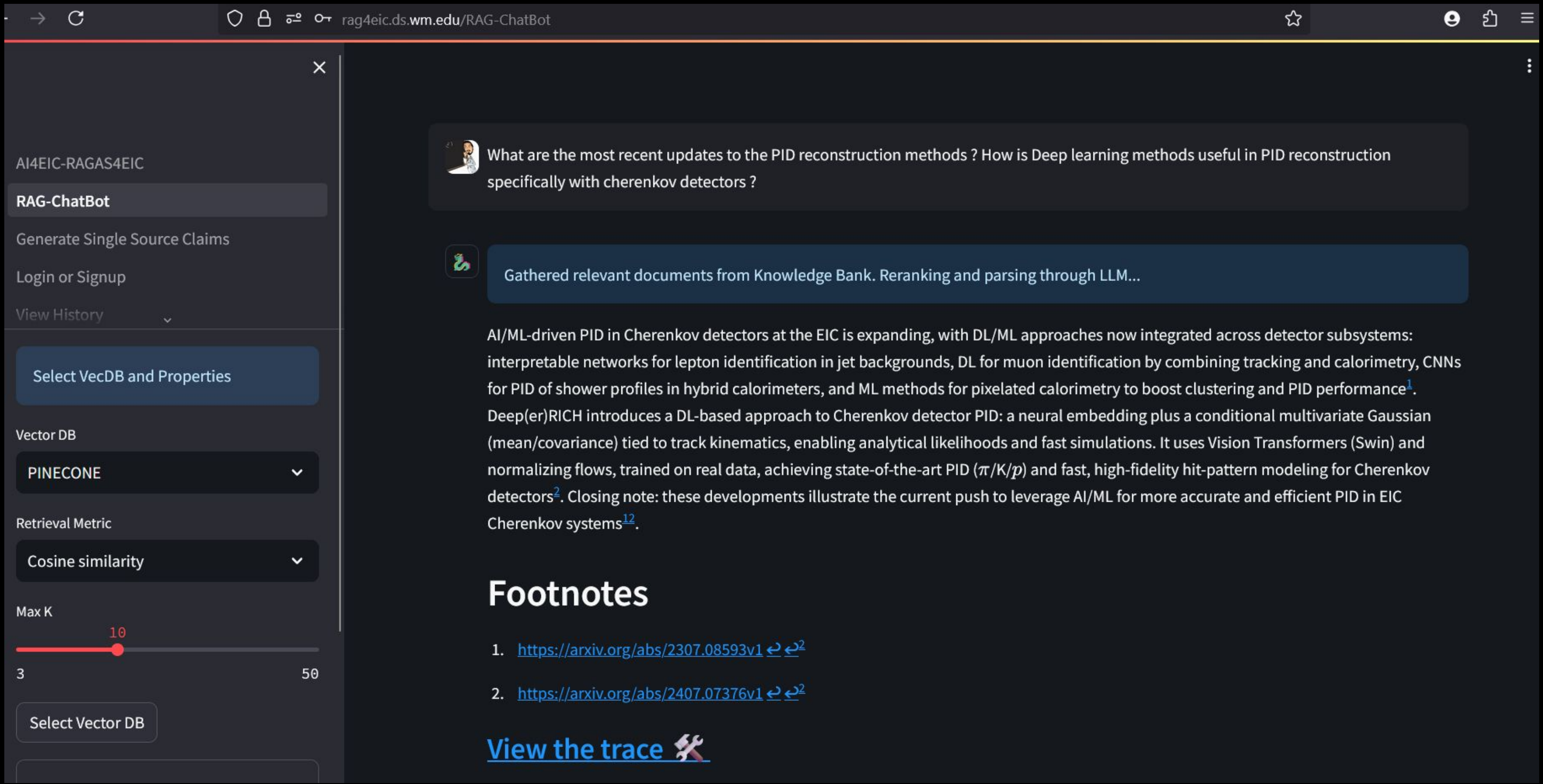


The Generic Workflow of building Traditional RAG

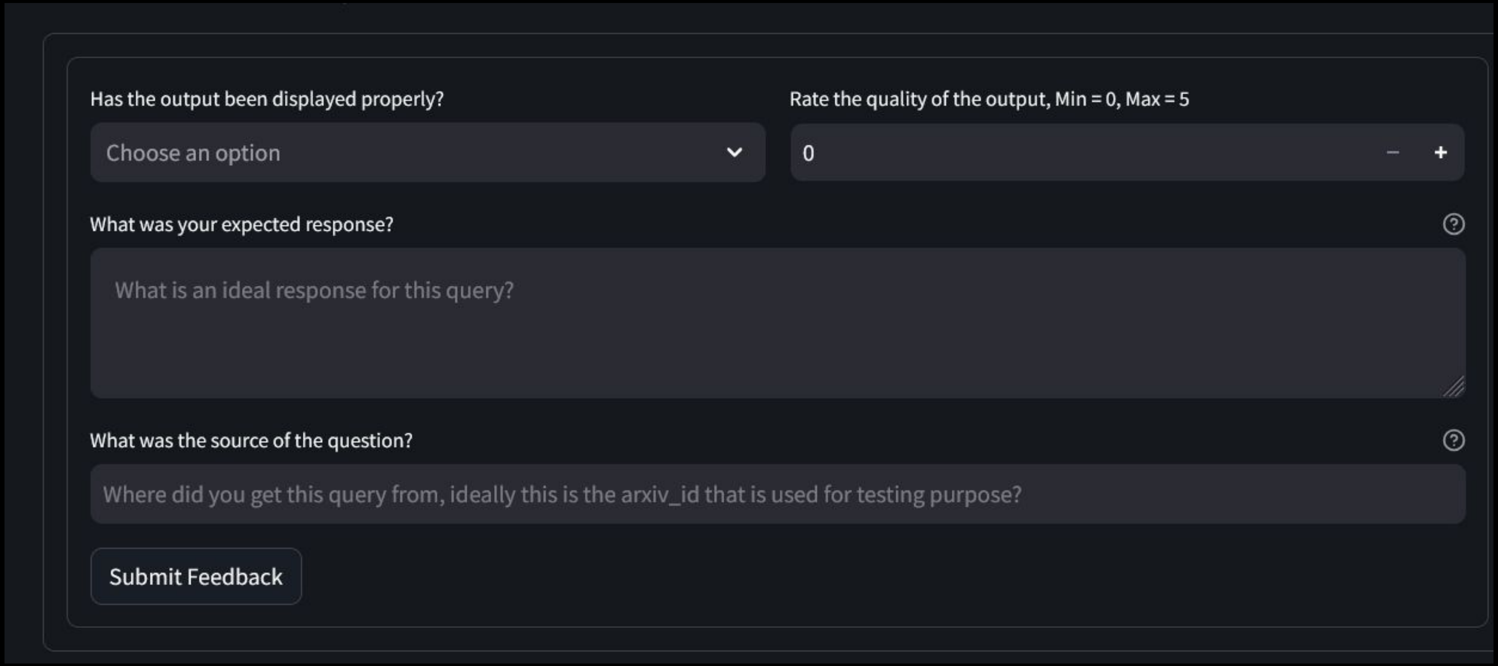


A typical RAG application

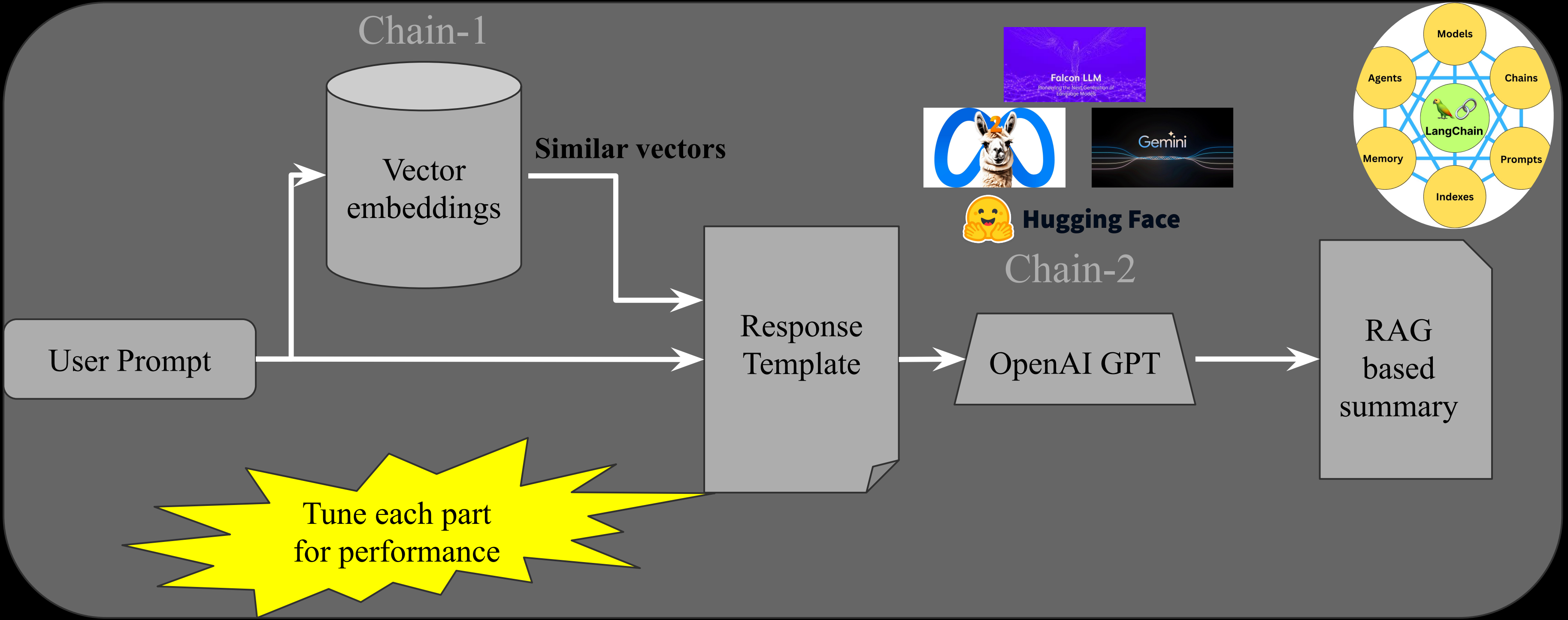
Check out the full trace of the pipeline in langsmith



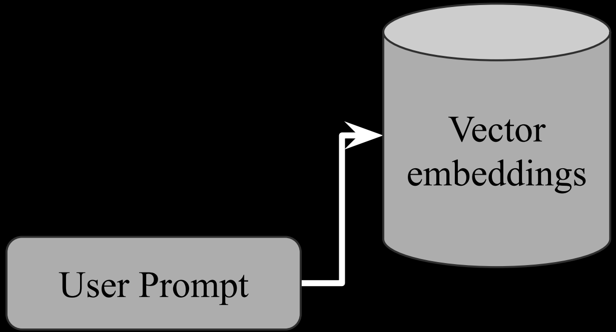
A typical RAG application – Feedback for fine tuning later

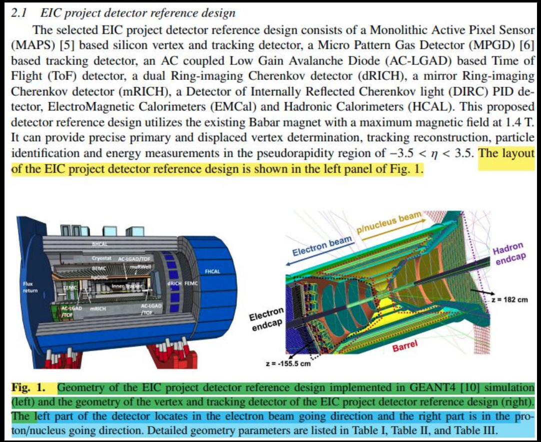


The Generic Workflow of building Traditional RAG



Ingestion tuning – LateX vs PyPDF Splitter





2.1 EIC project detector reference design

The selected EIC project detector reference design consists of a Monolithic Active Pixel Sensor (MAPS) [5] based silicon vertex and tracking detector, a Micro Pattern Gas Detector (MPGD) [6] based tracking detector, an AC coupled Low Gain Avalanche Diode (AC-LGAD) based Time of Flight (ToF) detector, a dual Ring-imaging Cherenkov detector (dRICH), a mirror Ring-imaging Cherenkov detector (mRICH), a Detector of Internally Reflected Cherenkov light (DIRC) PID detector, ElectroMagnetic Calorimeters (EMCal) and Hadronic Calorimeters (HCAL). This proposed detector reference design utilizes the existing Babar magnet with a maximum magnetic field at 1.4 T. It can provide precise primary and displaced vertex determination, tracking reconstruction, particle identification and energy measurements in the pseudorapidity region of $-3.5 < \eta < 3.5$. The layout of the EIC project detector reference design is shown in the left panel of Fig. 1.

Fig. 1. Geometry of the EIC project detector reference design implemented in GEANT4 [10] simulation (left) and the geometry of the vertex and tracking detector of the EIC project detector reference design (right). The left part of the detector locates in the electron beam going direction and the right part is in the proton/nucleus going direction. Detailed geometry parameters are listed in Table I, Table II, and Table III.

2305.15593v1

Similarity index

Question 1: Where will EIC experiment be built?

LateX Splitter – 0.37; PyPDF Splitter – 0.32

Question 2: How are dRICH detectors optimized at EIC?

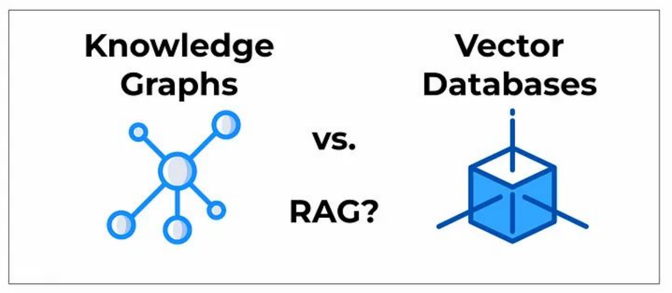
LateX Splitter – 0.26; PyPDF Splitter – 0.21

Question 3: Give me the latest update on EIC tracker

LateX Splitter - 0.35, PyPDF Splitter – 0.29

- PyPDF Splitter splits based on text only, while LateX splitted splits based on “tags”
- Length of each
- Figure caption can be tagged separately

Retrieval tuning – Similarity searches



Traditional Vector DB –

- uses similarity indices to retrieve information
- k-Nearest Neighbours
- Maximal Marginal Relevance – reduces redundant phrases

Graph Vector DB -

- Uses Knowledge graph for better contextual retrieval
- Saves information as a node in a graph with connections defining the strength to other “words”/phrases

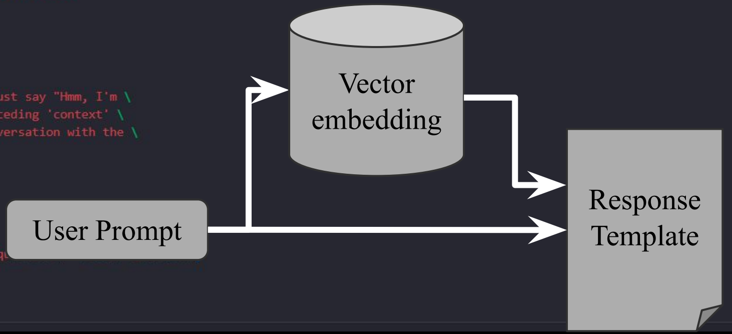
Context tuning

Setting the appropriate context with LLM.

Crucial in getting desired outcome.

Ongoing efforts to come up with Zero Shot Prompt for the summarization task.

```python
from langchain.prompts import PromptTemplate

template2 = """\
You are an expert in providing up to date information about the Electron Ion Collider (EIC), tasked with answering any question \
about EIC based only on the provided context.

Generate a comprehensive, and informative answer of 100 words or less for the \
given question based solely on the provided search results (URL and content). You must \
only use information from the provided search results. Use an unbiased and \
journalistic tone. Combine search results together into a coherent answer. Do not \
repeat text. You should use bullet points in your answer for readability. Make sure to break down your answer into bullet points.\
You should not hallicunate nor build up any references, Use only the `context` html block below and its associated `ARXIV_ID` if you find the context relevant.
Make sure not to repeat the same context. Be specific to the exact question asked for.\
After each bullet point, cite up to 5 most relavant arxiv_id associated with the `context` html block from which the bullet point was generated. \
The citations should be taken from the context between the tags <ARXIV_ID> and <ARXIV_ID/>. Only quote relavant arxiv_id \

Only quotee the most relevant arxiv_id that you find from `context` block. Note that the contexts are numbered according to the cosine similarity index.\
Place these citations at the end of the sentence or paragraph that reference them. If different results refer to different entities within the same name, write separate \
answers for each entity.

If there is nothing in the context relevant to the question at hand, just say "Hmm, \
I'm not sure." Don't try to make up an answer. Write the answer in the form of markdown bullet points.\
Make sure to highlight the most important key words in red color. Be precise and concise.\

Anything between the following `context`  html blocks is retrieved from a knowledge \
bank, not part of the conversation with the user. The context are numbered based on its knowledge retrival and increasing cosine similarity index. \
After each context the arxiv_id is given within the <ARXIV_ID> html block. \
Make sure to consider the order in which they appear context appear. It is an increasing order of cosine similarity index.\
The contents are formatted in latex, you need to remove any special characters and latex formatting before cohercing the points to build your answer.\
You will cite no more than 5 citations from the context below.\
Make sure these citations have to relavant as well as not repetitive in nature.

<context>
    {context} <ARXIV_ID> {arxiv_id} <ARXIV_ID/>
<context/>

REMEMBER: If there is no relevant information within the context, just say "Hmm, I'm \
not sure." Don't try to make up an answer. Anything between the preceding 'context' \
html blocks is retrieved from a knowledge bank, not part of the conversation with the \
user.\
Question: {question}
"""
rag_prompt_custom = PromptTemplate.from_template(template2)

print (rag_prompt_custom.invoke(
    {"context": "filler context", "arxiv_id" : "filler arxiv_id", "q
    ).to_string()
)
```
✓ 0.0s



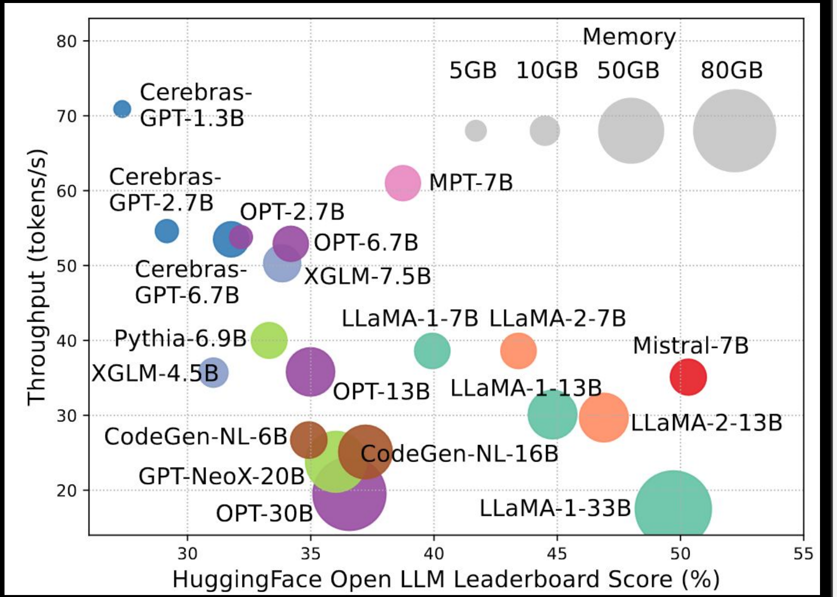
Checkout the response templates (ai4eic github link)

The LLM Model

Choice of LLM model as well affects the performance of the RAG system.

Small/Medium Language Models O(10B) OpenLLM models are available

Need to consider Model complexity (resource utilization) vs performance.



arxiv:2312.03863

Limitations of Standard RAG

- Shallow Retrieval: Embeddings may miss context or multi-hop reasoning
- Static Retrieval pipeline: No dynamic decision making on what and how to retrieve
- Context stuffing: Retrieved chunks are just concatenated — can cause token overload or irrelevant context.
- Lost in the middle: Classic case of transformer's attention mechanism limitation. Content in the middle of the context is usually less attended to. 2307.03172
- Prompt Injection: In RAG, the retrieved text (from documents, DBs, or the web) is directly injected into the LLM's prompt as context. This is also true for other forms of RAG (if not implemented with caution)

Prompt injection via screenshots in Perplexity Comet

3. The RAG Approach

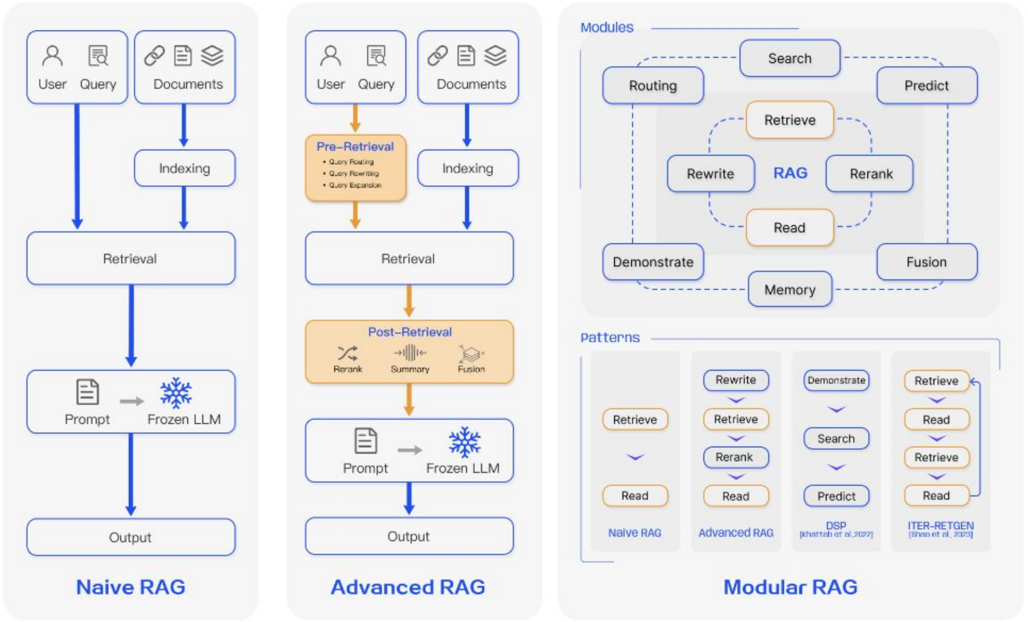


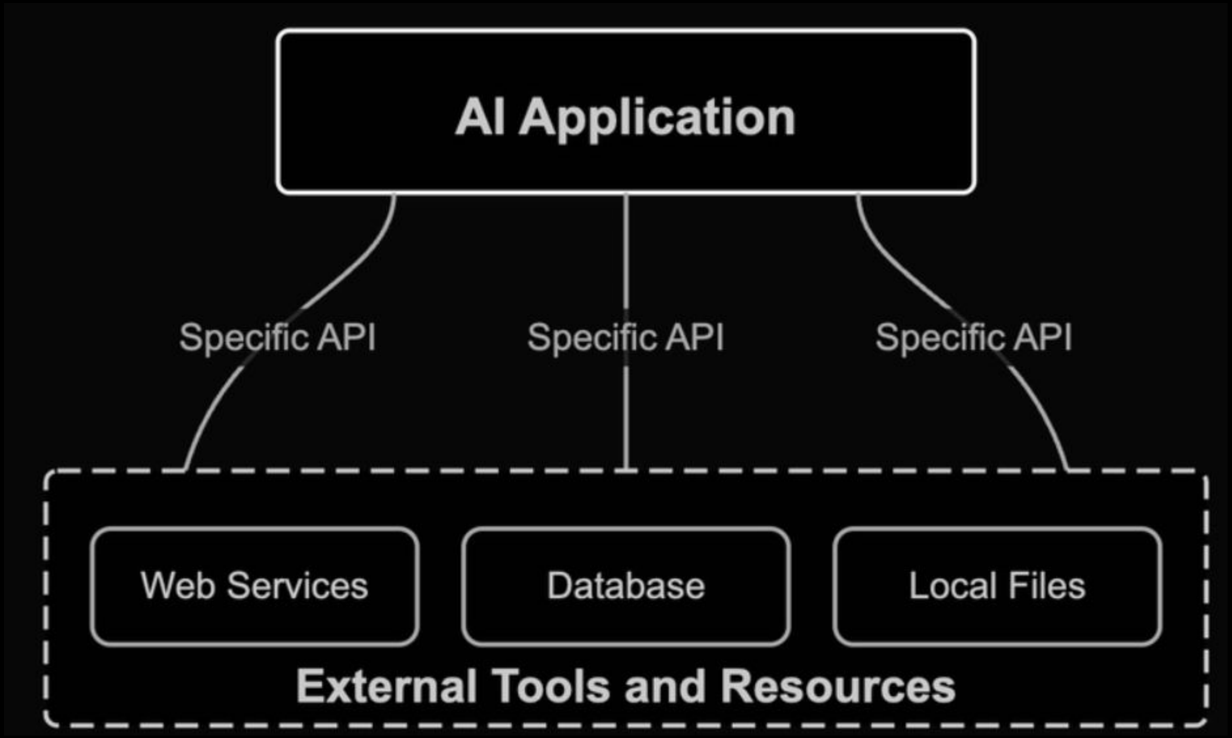
Figure 3: Comparison between the three paradigms of RAG

arxiv:2312.10997

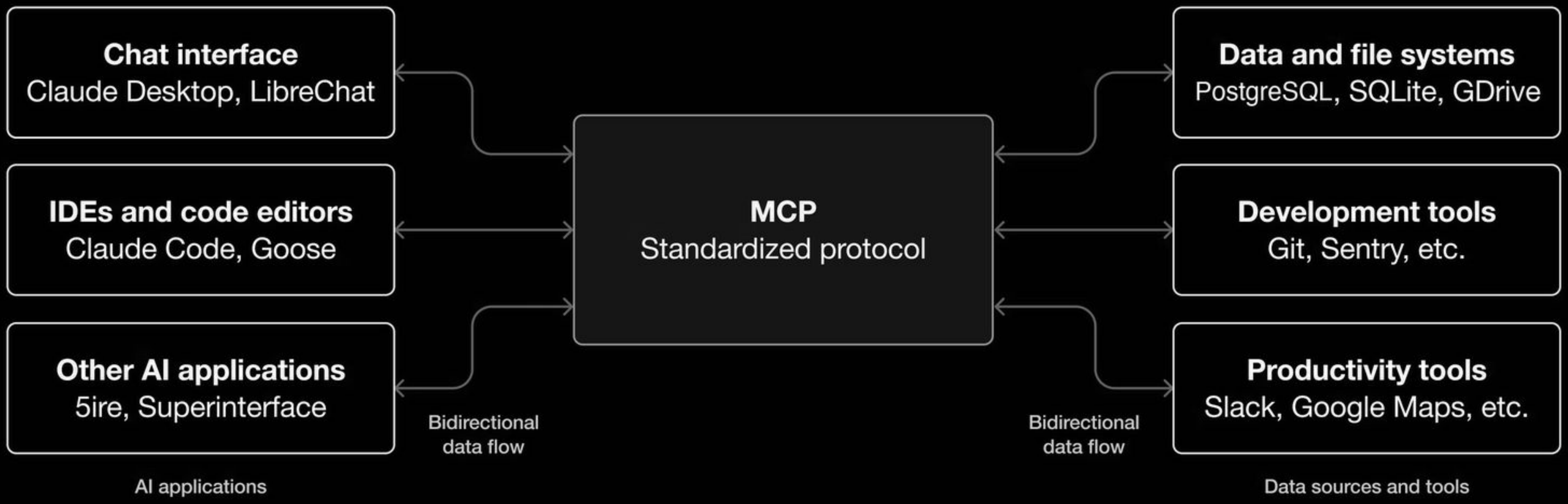
Advanced RAG:
Using LLM more than a orchestrator. Eg. CoT based RAG

Modular RAG:
Using LLM for routing logic. Give access to tools like TWikiReader to perform actions based on results from LLM.

The problem with Orchestration



3. Model Context Protocol modelcontextprotocol.io



Evaluating RAG:
A targeted example

How to evaluate this response?

Ragas score

1. Faithfulness –

$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}$$

2. Answer relevance –

$$\text{Answer relevance} = \frac{|\text{\#claims correctly responded}|}{|\text{\#claims in given text}|}$$

3. Context relevance –

$$\text{Context relevance} = \frac{|\text{\# relevant unique refs}|}{|\text{\# refs retrieved}|}$$

4. Answer correctness – Qualitative. Need ground truth.

Question 3: What is the pixel size and expected momentum resolution at 2GeV from a MAPS ITS3 tracking layer

Iteration 1

- The pixel size of the MAPS ITS3 tracking layer is 10 μm
  <ARXIV_ID>http://arxiv.org/abs/2108.08314v1</ARXIV_ID> https://arxiv.org/abs/2108.08314v1
  <ARXIV_ID>http://arxiv.org/abs/2203.07626v2</ARXIV_ID>
  <ARXIV_ID>http://arxiv.org/abs/2302.01447v1</ARXIV_ID>
  <ARXIV_ID>http://arxiv.org/abs/2305.15593v1</ARXIV_ID>
  <ARXIV_ID>http://arxiv.org/abs/2210.08609v2</ARXIV_ID>
- The expected momentum resolution at 2 GeV from a MAPS ITS3 tracking layer is 2.5% to 15% of the initial momentum <ARXIV_ID>http://arxiv.org/abs/2305.15593v1</ARXIV_ID>
  <ARXIV_ID>http://arxiv.org/abs/2210.08609v2</ARXIV_ID> https://arxiv.org/abs/2210.08609v2

Evaluating RAG: Another more complex full example

How to evaluate this response?

Ragas score

1. Faithfulness ~ 4/25
2. Answer relevance – 1
3. Context relevance – 1
4. Answer correctness – Qualitative

- Targeted questions are relatively easier to quantify and evaluate.
- Hence, Classic Metrics are metric that require human expertise while RAGAS metric can be automated using LLM

Question 3: What are the dimensions of MAPS detector used in EIC?

Response

- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with a sensor size of 10x10 cm <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2302.01447v1', 'http://arxiv.org/abs/2210.04879v2', 'http://arxiv.org/abs/2305.02079v2', 'http://arxiv.org/abs/2302.01447v1']
- The ePIC MAPS electron endcap detector has a geometry parameter listed, and the ePIC MAPS hadron endcap detector also has geometry parameters listed <ARXIV_ID> ['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1']
- The ePIC MAPS vertex detector design can reach as low as 0.05, and the imaging layers of MAPS sensors provide excellent position resolution <ARXIV_ID> ['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2305.15593v1']
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']

https://arxiv.org/abs/2203.07626v2

https://arxiv.org/abs/2302.01447v1

https://arxiv.org/abs/2210.04879v2

https://arxiv.org/abs/2305.02079v2

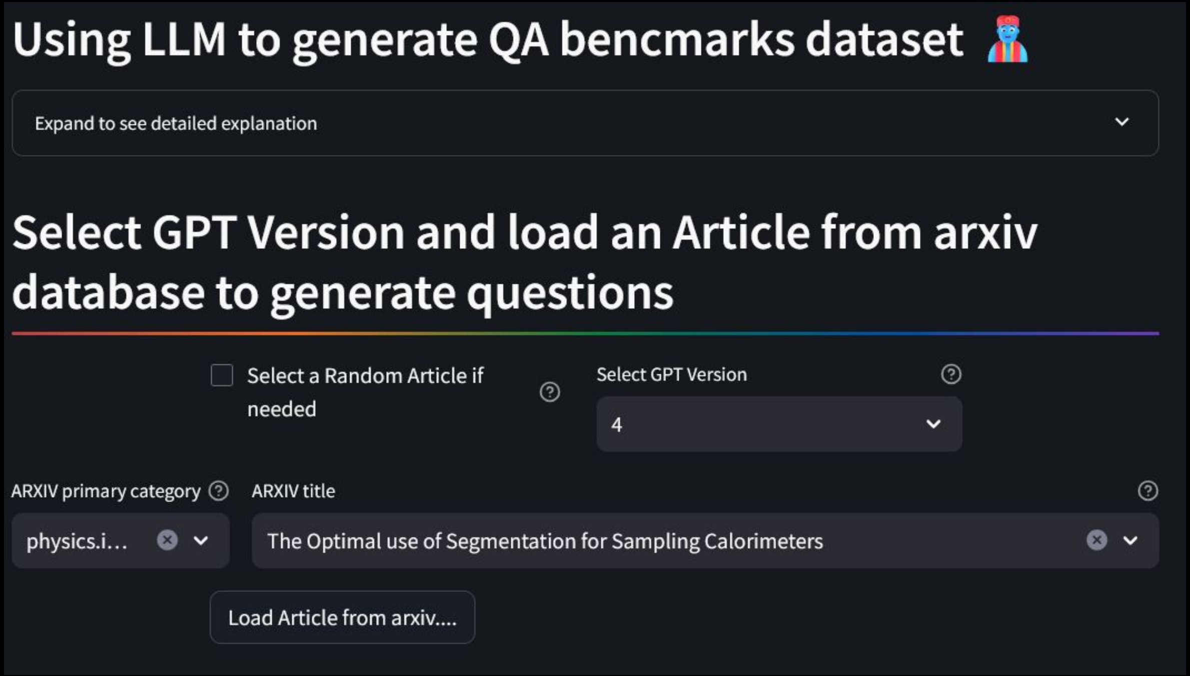
https://arxiv.org/abs/2305.15593v1

But can we use a LLM for QA Generation?

Use LLM for QA Generation. [Set context for generating questions](#).

Select an arxiv article. From the database

Specify the number of claims to generate, along with expected answers.



[Check out the trace here](#)

But can we rely on LLM for QA Generation?

Using a more complex LLM can generate questions along with answers.

LLM can automate the generation of questions to specific format which can be mixed to form complex questions.

[A small sample dataset from 2023]

Question 1 from 2310.04442v1 at [http://arxiv.org/pdf/2310.04442v1]

Q: What are the sampling fractions for the HCAL and ECAL as computed using a 40 GeV electron? How is the reconstructed energy (EReco) calculated using these sampling fractions? A:


```
{
    "n_claims" : 2,
    "claims": [
        """sampling fractions for 'HCAL' and 'ECAL'""",
        """calculation of 'EReco' using sampling fractions"""
    ],
    "complete_response": """ The sampling fractions for the HCAL and ECAL are 2.2% and 3.0%, respectively.
                            The reconstructed energy (EReco) is calculated as the sum of the hit energy divided by th
                            expressed as EReco = (Sum of Ei in HCAL / SFhcal) + (Sum of Ei in ECAL / SFecal), where S
                        """,
    "answers": [
        """2.2% for HCAL and 3.0% for ECAL""",
        """EReco = (Sum of Ei in HCAL / 2.2%) + (Sum of Ei in ECAL / 3.0%)"""
    ]
}
```


[Check out the trace here]

Current efforts – Serve EIC community

- **Data Ingestion**
  - Ingest a large corpus of EIC documents into the current RAG pipeline
  - Use existing cloud-based vector store (e.g., Pinecone)
  - Integrate updated GPT-based models to serve initial responses
- **Beta User Rollout**
  - Launch access to ~100 beta users from the EIC community
  - Users are expected to:
    - Test the app regularly and give feedback on UI rendering
    - Evaluate retrieval quality and grounding of responses
    - Rate model answers using RAGAS-style LLM-as-judge scoring
  - Update to app Every 4 weeks:
    - App updates with improved capabilities and bug fixes
    - New content ingestion or UX enhancements

- **Goal:**
  - Evaluate various RAG strategies
  - Develop various ingestion strategies
  - Build a high-quality, supervised dataset for fine-tuning
- **Cloud resource support:**
  - Provided by W&M for AI4EIC

*Upto 8 months

Serving the EIC community – Beta users

- Cloud-based deployment planned
  - To enable scalable access for beta testers and ensure smooth delivery of RAG4EIC.
- Corpus size and system load increasing
  - As we ingest more EIC related documentation, self hosting becomes less feasible* without dedicated infrastructure.
- Need for robust model performance
  - Open-source small language models, without fine-tuning, struggle to meet quality benchmarks like **RAGAS**
- Dedicated compute required
  - Hosting large models or experimentation with fine-tuned LLMs (e.g., LLaMA, Mistral, Phi) demands consistent cloud-backed resources.
- Supports iterative dataset creation
  - Beta user queries and usage logs will help build a real-world EIC-focused dataset to improve RAG accuracy and utility.
- Laying the groundwork for model fine-tuning
  - Early user interactions help us gather a high-quality dataset for future domain-specific training.

*currently with traditional RAG

Serving the EIC community – Modularized implementation

- MCP Integration
  - Implement data sources as MCP servers (same corpus as cloud)
  - Mainly to alleviate privacy concerns
  - Develop batteries to run model inference + retrieval via MCP
  - Focus: Internal/private hosting of RAG workflows (e.g., BNL, JLab)
  - Role based authentication for VectorDB
- LangGraph Agentic Pipeline
  - Replace LangChain with LangGraph for modular graph-based control.
  - Introduce:
    - Source-aware routing
    - Multi-hop retrieval
    - Agentic scoring or fallback logic
  - Improved orchestration and auditability of queries
- Unified Deployment
  - Combine all in one stack to be deployed in a site

*currently with traditional RAG

List of tasks / areas of involvement

- Towards serving the first version to beta users
    - Ingestion
        - Arxiv papers ✅
        - Indico meetings page ✋
    - Inference
        - Add conversational memory
        - Add support for multiple vector base calls ✅
    - Web interface
        - Improve feedback mechanism ✅

- Modular implementation
    - Agentic workflow
        - Replace LangChain with LangGraph:v1 🔥
        - Supervisor Agent
        - Network agent implementation
    - Algorithm
        - Advanced RAG ✅
        - Graph RAG
        - Evaluation of RAG pipeline
    - MCP server implementation
        - Wiki sources
        - Zenodo sources ✅
    - Web application development
        - Implementation of LibreChat interface ✋ 🔥

Ofcourse, new issues are welcome

How to get involved

- Email to support@eic.ai
- Subject line: Involvement in RAG4EIC as developer (or beta user)
- If developer (highly recommend including)
  - Current institution and a brief experience with LLM development (Just to get to know)
  - A brief description on which of the areas you would like to work on
- If beta user (highly recommend including)
  - Current institution and a brief description on how you are currently using Language model in your research
  - Your area of expertise (Eg. Theory, experimental, hadron spectroscopy)
  - Would you be interested in curating a golden data set in your area of expertise
- Regular RAG4EIC working group meetings
  - Once a month, Next meeting By August 26 2025 (Tuesday) anticipated. A reminder will follow.

Backups

A full example

```python
ques = r"What are the dimensions of MAPS detector used in EIC?"
print ("# Question 3: " + ques)
for i in range(5):
    print (f"## Iteration {i+1}")
    for chunk in rag_chain_with_source.stream(ques):
        if(chunk.get("answer")):
            print(chunk["answer"], end="", flush=True)
    print ("\n")
```

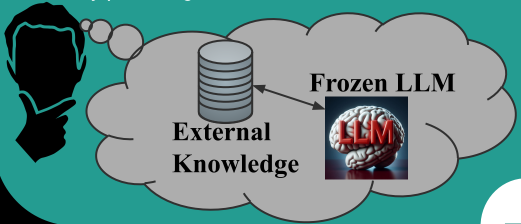
✓ 28.6s Python

```
# Question 3: What are the dimensions of MAPS detector used in EIC?
## Iteration 1
- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with a sensor size of 10x10 cm, and a length of 2.4 m. The detector consists of 5 layers and utilizes (

## Iteration 2
- The MAPS detector used in the Electron Ion Collider (EIC) has dimensions ranging from 3.67 cm to 43 cm, with the electron endcap detector having a geometry of 23 cm to 43 cm. The dete

## Iteration 3
- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with various inner and outer radii for different components of the detector. The ePIC MAPS electron enc

## Iteration 4
- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with a sensor size of 10x10 cm <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2
- The ePIC MAPS electron endcap detector has a geometry parameter listed, and the ePIC MAPS hadron endcap detector also has geometry parameters listed <ARXIV_ID>['http://arxiv.org/abs/
- The ePIC MAPS vertex detector design can reach as low as 0.05, and the imaging layers of MAPS sensors provide excellent position resolution <ARXIV_ID>['http://arxiv.org/abs/2203.0762
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']

## Iteration 5
- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with a sensor size of 10x10 cm, and a length of 2.4 m. The detector also features a 65 nm Monolithic Ac
```

RAG based summarization AI for Electron Ion Collider (EIC)

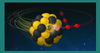
What is RAG ?

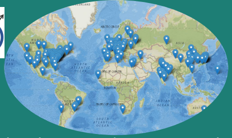
- Retrieval Augmented Generation (RAG).
- Access up to date information without explicitly training of LLM.
- Reduce “Hallucination” of LLM.
- Grounding LLM to truth to increase reliability by providing citations.



Why need RAG for Large Scale Physics Experiments?

- Electron Ion Collider (EIC) is a large scale experiment.
- Regular updates to documents, Run Wiki
- Newbies may take 6 months to get to know the full experimental details.
- Document size ∝ Scale of experiment

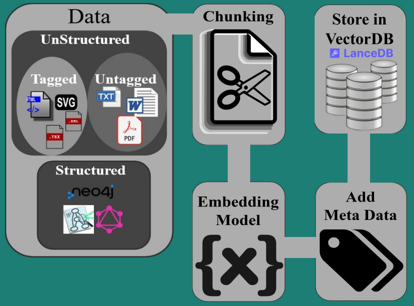




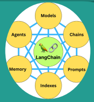
1400 physicists, 240 institutions and 38 countries

Ingestion

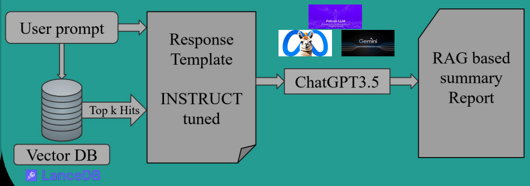
- Creation of the vectorized knowledge base.
- Every node below influence RAG performance
- 200 recent arxiv papers on EIC (since 2021)



The inference*



- Given a prompt compute similarity index to most similar vectors in VectorDB
- Use LLM to further narrow down and summarize the finding



*Naive RAG pipeline

The research approach

What we have built ?

Question 3: What is the pixel size and expected momentum resolution at 2GeV from a MAPS ITS3 tracking layer

Iteration 1

- The pixel size of the MAPS ITS3 tracking layer is 10 μm <ARXIV_ID>http://arxiv.org/abs/2108.08314v1</ARXIV_ID> <ARXIV_ID>http://arxiv.org/abs/2203.07626v2</ARXIV_ID> <ARXIV_ID>http://arxiv.org/abs/2302.01447v1</ARXIV_ID> <ARXIV_ID>http://arxiv.org/abs/2305.15593v1</ARXIV_ID> <ARXIV_ID>http://arxiv.org/abs/2210.08609v2</ARXIV_ID>
- The expected momentum resolution at 2 GeV from a MAPS ITS3 tracking layer is 2.5% to 15% of the initial momentum <ARXIV_ID>http://arxiv.org/abs/2305.15593v1</ARXIV_ID> <ARXIV_ID>http://arxiv.org/abs/2210.08609v2</ARXIV_ID>

Question 3: What are the dimensions of MAPS detector used in EIC?

Response

- The MAPS detector used in the EIC has dimensions ranging from 3.67 cm to 43 cm, with a sensor size of 10x10 cm <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2302.01447v1', 'http://arxiv.org/abs/2210.04879v2', 'http://arxiv.org/abs/2305.02079v2', 'http://arxiv.org/abs/2302.01447v1']
- The ePIC MAPS electron endcap detector has a geometry parameter listed, and the ePIC MAPS hadron endcap detector also has geometry parameters listed <ARXIV_ID> ['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1', 'http://arxiv.org/abs/2305.15593v1']
- The ePIC MAPS vertex detector design can reach as low as 0.05, and the imaging layers of MAPS sensors provide excellent position resolution <ARXIV_ID> ['http://arxiv.org/abs/2203.07626v2', 'http://arxiv.org/abs/2305.15593v1']
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']
- The MAPS sensor used in the EIC is based on low-mass, low-power consumption MAPS in 65 nm technology <ARXIV_ID>['http://arxiv.org/abs/2203.07626v2']

- Creation of benchmark evaluation dataset.
- Evaluate performance before scaling.

Methods to evaluate RAG’s performance

- RAGAS score
- LLM as Judge[1]

1. Faithfulness –

$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}$$

2. Answer relevance –

$$\text{Answer relevance} = \frac{|\text{\#claims correctly responded}|}{|\text{\#claims in given text}|}$$

3. Context relevance –

$$\text{Context relevance} = \frac{|\text{\# relevant unique refs}|}{|\text{\# refs retrieved}|}$$

Methods to improve RAG architecture

- Better chunking strategies. LateX Splitter, TWikiSplitter
- Metadata based filtering.
- Response Template fine-tuning. INSTRUCT tuning
- Model fine-tuning. Computationally costly.

Fine tuning