Retrieval Augmented Generation (RAG): Bridging Document Analysis and Recognition with Large Language Models

ICDAR, 2024

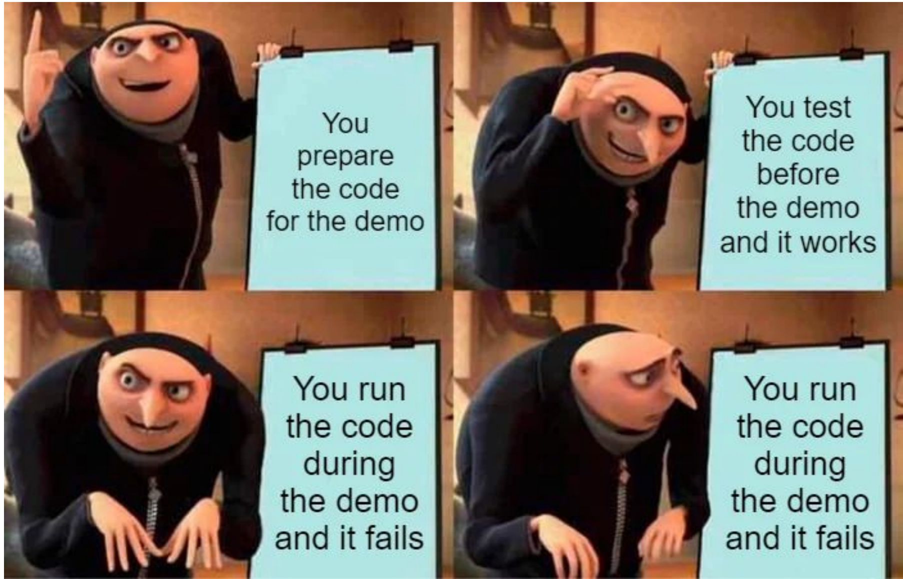
infocusp Innovations

Namastey

- From Ahmedabad, India
- Chief ML Officer at Infocusp
- 9+ YoE in ML/ DL/ LLMs
- Rajyoga meditation practitioner
- Calvin & hobbes fan :)

- Familiarity with python/ LLMs/ Llamaindex?
- Goal is to give pointers and a starting point
- Colabs included - run later
- Lessons from building LLM applications

Disclaimer: just in case :)



Outline:

Part I

- What is RAG?
- Real world case studies/ motivation
- RAG pipeline components
- Data preparation
- LLMs
- Low/ no code RAG solutions - building your first RAG application
- Limitations of RAG

Part II

- Embeddings
- Retrieval: Vector DB
- Retrieval: Distance metrics
- End to end RAG hands on using Langchain

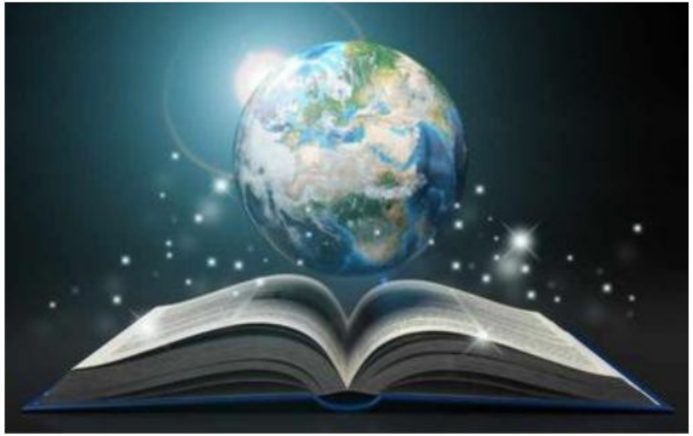
- Improving RAG: Reranking
- Improving RAG: Query rewriting
- Multi modal RAG
- Graph RAG
- RAG evaluation

What is RAG?

We all know what a RAG is, it does not warrant a tutorial at ICDAR :p

Limitations of LLM: world knowledge/ max size vs personal

Best of both: language of LLMs and knowledge of your documents (private + reliable)



infocusp Innovations

Grounding of answers: hands on

bit.ly/grounding-llms

Real world case studies

Review analysis

Chat about the survey findings

Find specific responses

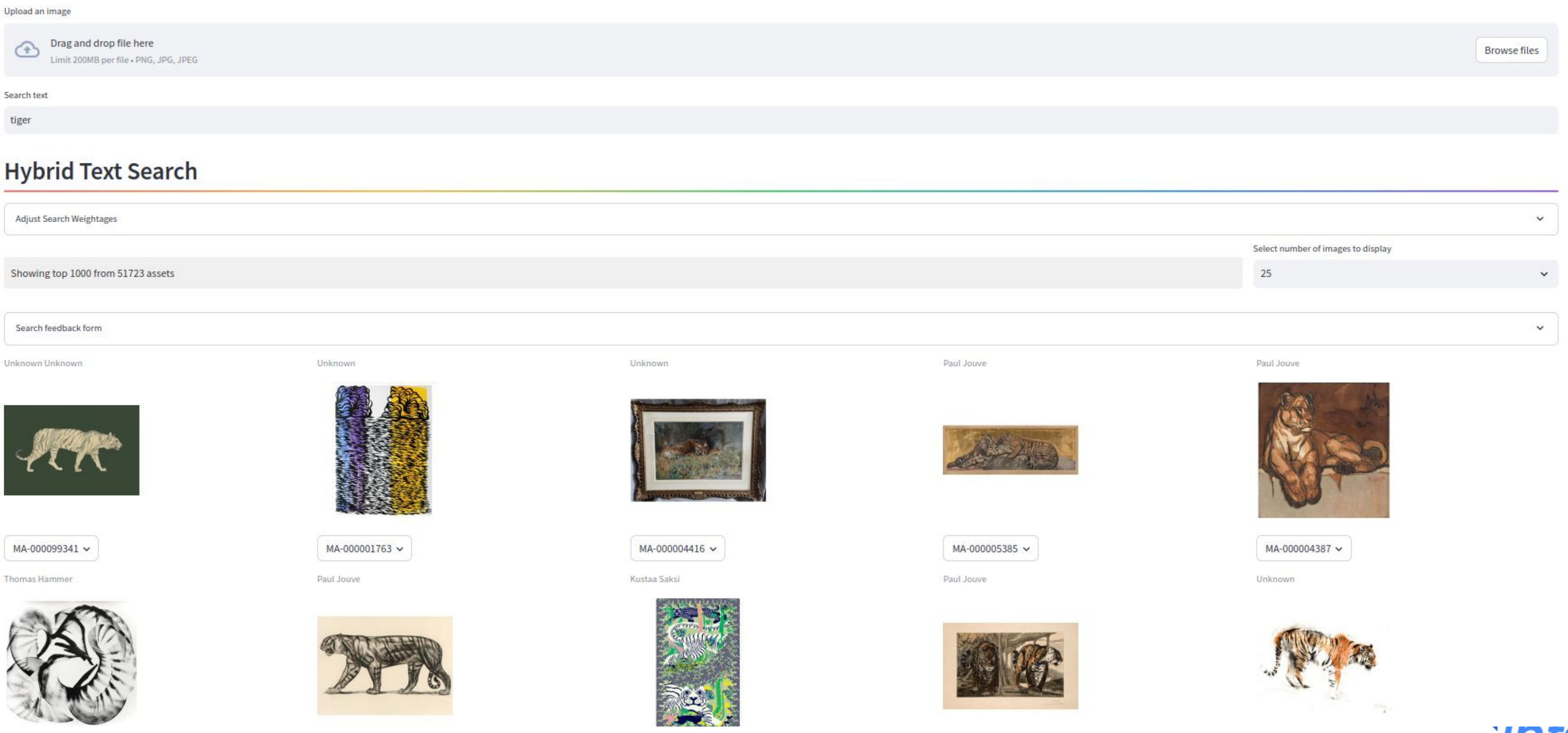
Real world case studies: Legal documents/ contracts

Finding relevant information

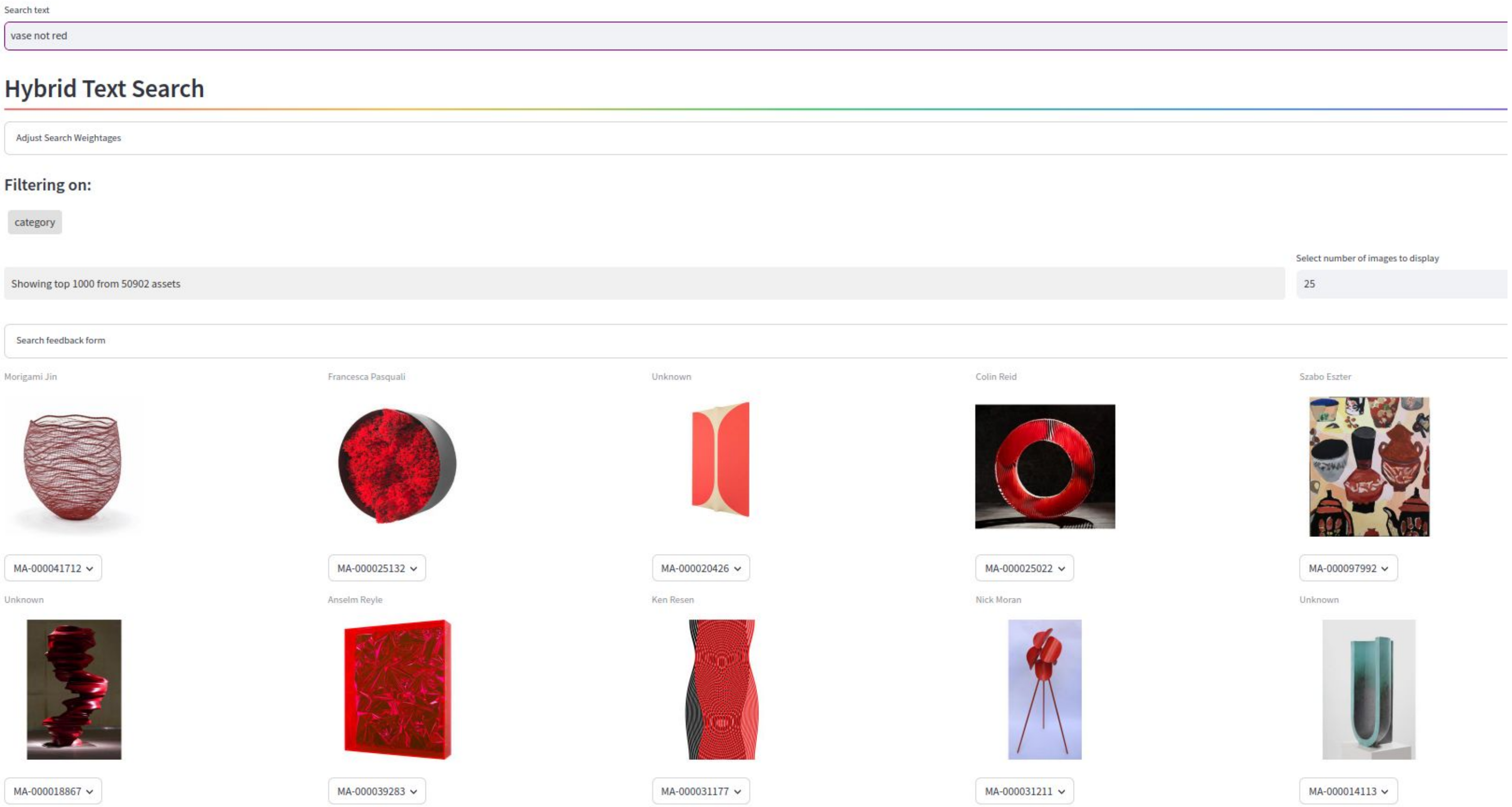
Browsing through hundreds of documents

Respond citing sources

Image search using text or images

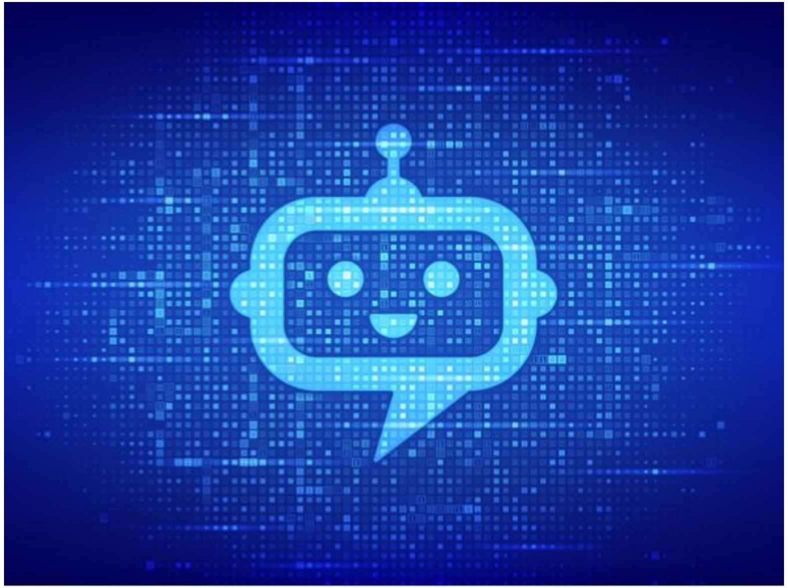


Smart but at the same time



Advanced/ research based use cases

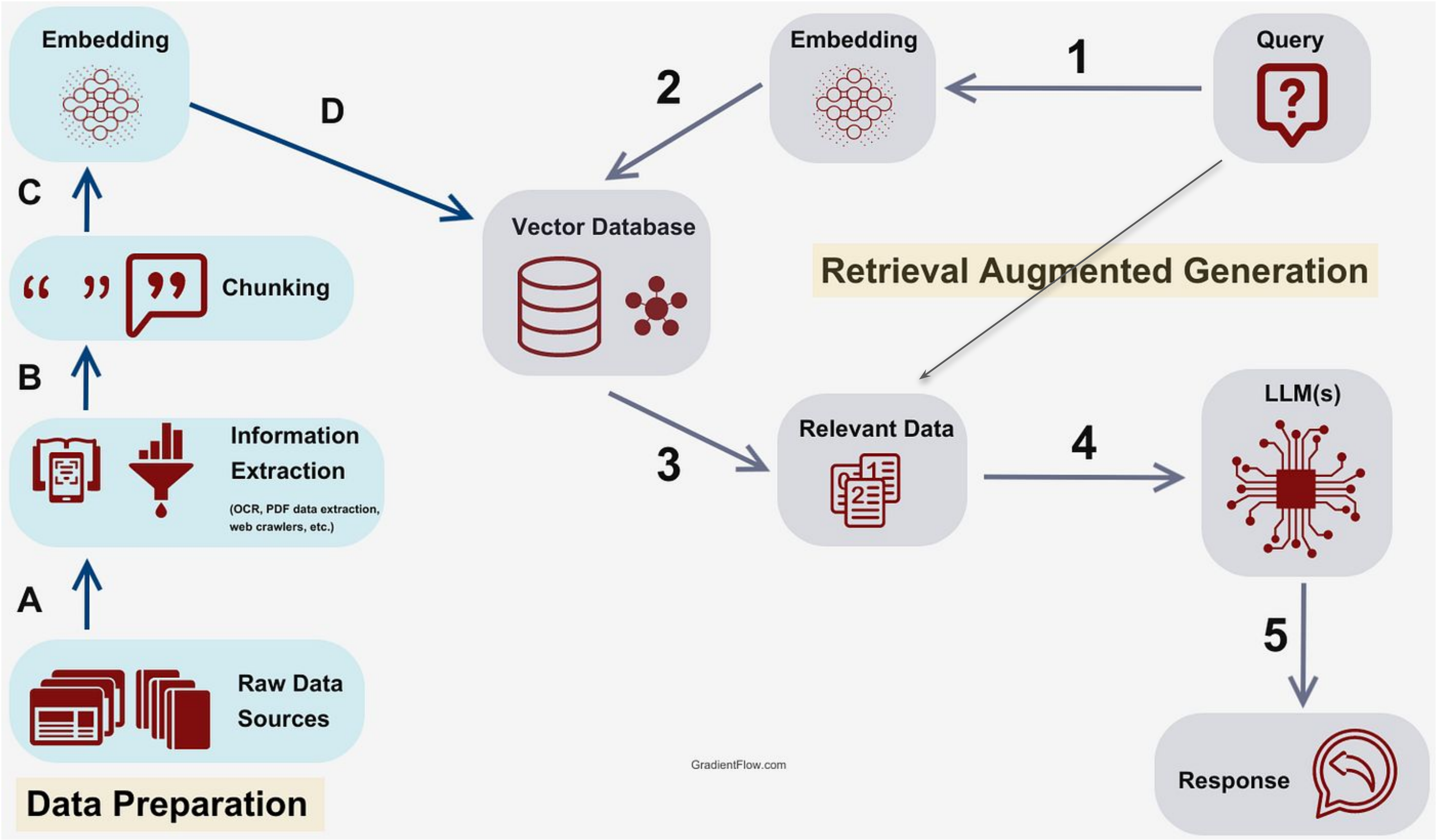
- Research assistant for materials discovery
- Browsing through a ton of marketing materials and summarizing



Limitations no one will tell you about

- Multi modal systems still do not understand negation (virtue of its training)
- Handling tabular data
- Holistic understanding of the subject and summarizing : although graph RAG handles this to a point
- Sensitivity to slight variations in prompts
- Not reproducible
- Ever updating black box models
- Hallucinations - we'll see tips to reduce this

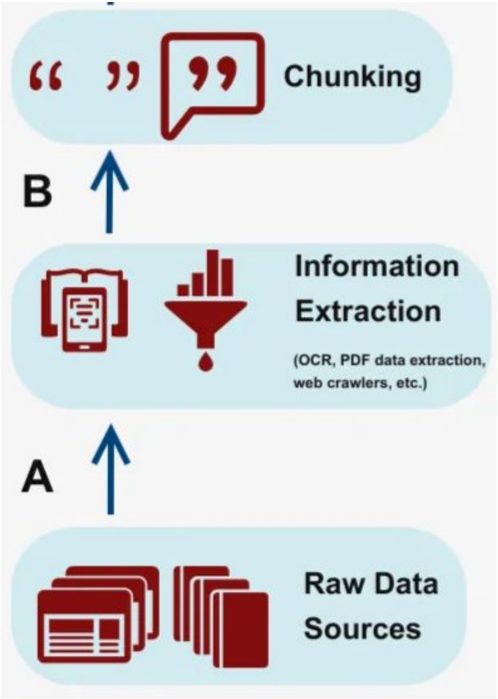
RAG pipeline



Source: https://medium.com/enterprise-rag/an-introduction-to-rag-and-simple-complex-rag-9c3aa9bd017b

No code app 1 hands on: Tour planner chatbot for Greece

Data Preparation



Hard facts

- Data will almost never be clean
- If it was so simple to load and analyze, someone would have done it already
- Each author has their own format
- Data cleaning forms a major part of any data driven endeavour

Data preparation

- Data loaders for various formats are available: html/ json/ txt/ code!
- Variety of data connectors available
- Chunking by
  - Splitting into fixed sized sentences (Cheap and simple)
  - Recursive splitting- we'll check it out later
  - Semantic chunking (Cost considerations)

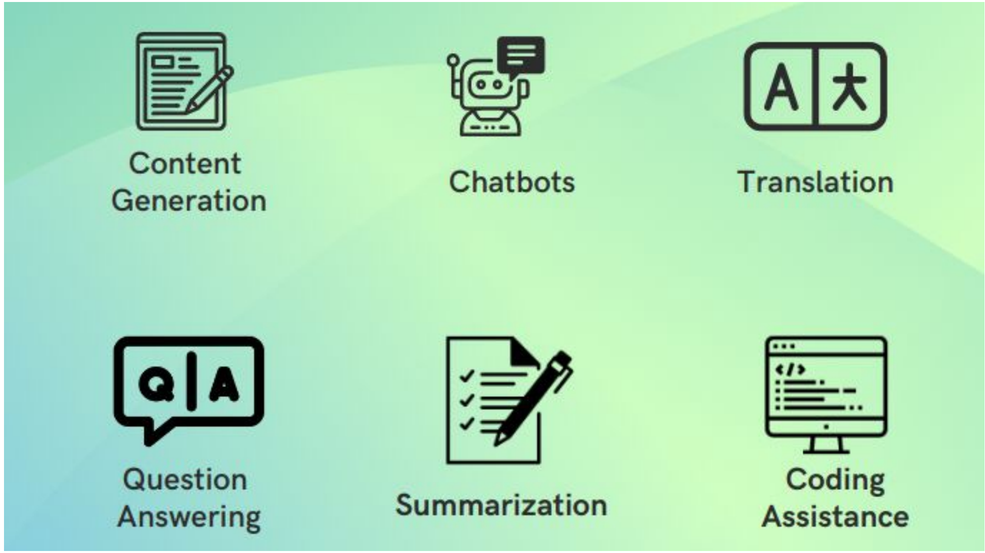
Hands on colab: data loading/ chunking

https://bit.ly/data-chunk

Large Language Models

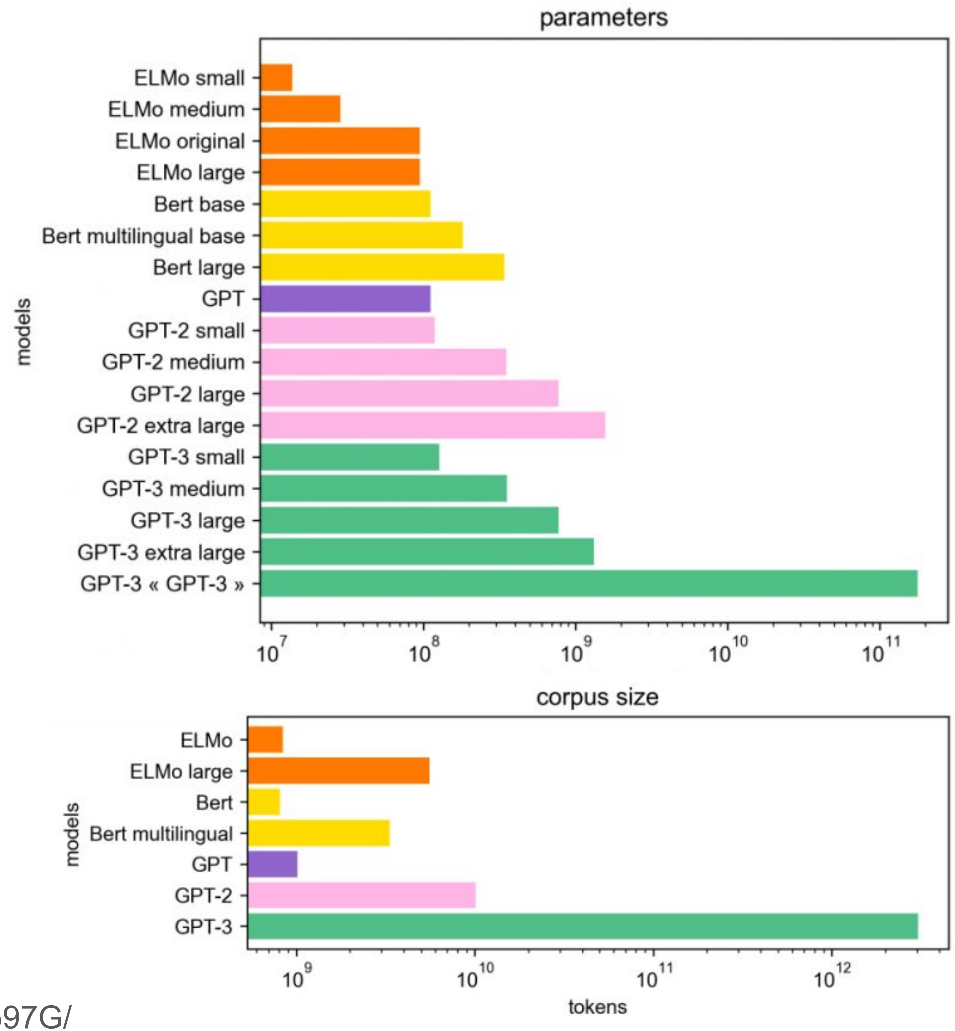
Large Language Models

- The backbone of everything that we will build
- Have some amazing capabilities virtue of how they're trained
- Could we train/ host our own LLMs?
- How did they get all these abilities?



Progress Highlights

- 2 Trillion parameters and growing
- 1 Trillion tokens and growing
- From LSTMs to transformers to MoE
- Next token prediction to MLM to NSP
- Fine tuning to few shot to one shot to zero shot



Ref: https://www.cs.princeton.edu/courses/archive/fall22/cos597G/

LLM in RAG

- 2 components would use it: Embedding generation and answering questions.
- Particularly more important when answering questions based on context

Questions/ doubts/ coffee?

Embeddings

- Literal: Convert from image/text/audio into a list of numbers. 🖼️ or 📄 => [1.2, 2.1, ...]. This process makes documents "understandable" to a machine learning model.
- By analogy: An embedding represents the essence of a document.
- Technical: Latent-space position of a document at a layer of a deep neural network.
- A small example: If you search your photos for "famous bridge in San Francisco". 🙂

Where are embeddings used

- Search (where text/ image results are ranked by relevance to a query string)
- Clustering (where text/ images are grouped by similarity)
- Recommendations (where items with related text strings are recommended)
- Anomaly detection (where outliers with little relatedness are identified)
- Diversity measurement (where similarity distributions are analyzed)
- Classification (where items are classified by their most similar label)

Embeddings progression

- Bag of words/ Count vectorizer
- TF-IDF
- Word2vec (CBOW or skipgram)
- Glove - co-occurrence probability prediction directly
- Contextual
  - BERT
  - LLMs

Compare embeddings

https://huggingface.co/spaces/mteb/leaderboard

Normalized Discounted Cumulative Gain for ranking
https://www.evidentlyai.com/ranking-metrics/ndcg-metric

Embeddings hands on: playing with glove

https://bit.ly/icdar-embeddings

Vector Databases

Vector DB considerations in industry

- Data management: Data storage, like inserting, deleting, and updating data.
- Metadata storage and filtering: Store metadata associated with each vector entry and filter based on that. Example: 🙂
- Hybridization: Combine with search
- Scalability: Scale with growing data volumes and user demands, providing better support for distributed and parallel processing
- Real-time updates: Vector databases often support real-time data updates, allowing for dynamic changes to the data to keep results fresh

infocusp Innovations

VectorDB considerations in the industry

- Ecosystem integration: Integrate with other components of a data processing ecosystem
- Data security and access control: Data security features and access control mechanisms to protect sensitive information
- Functionalities provided by the vectorDB off the shelf
- Pricing!!

Vector Stores

- Available options: https://docs.llamaindex.ai/en/stable/module_guides/storing/vector_stores/
- Simple for prototyping: ChromaDB
- Production Grade: Pinecone/ VertexAI Datastore/ AlloyDB
- Customizable: ElasticDB

Hands on Pinecone

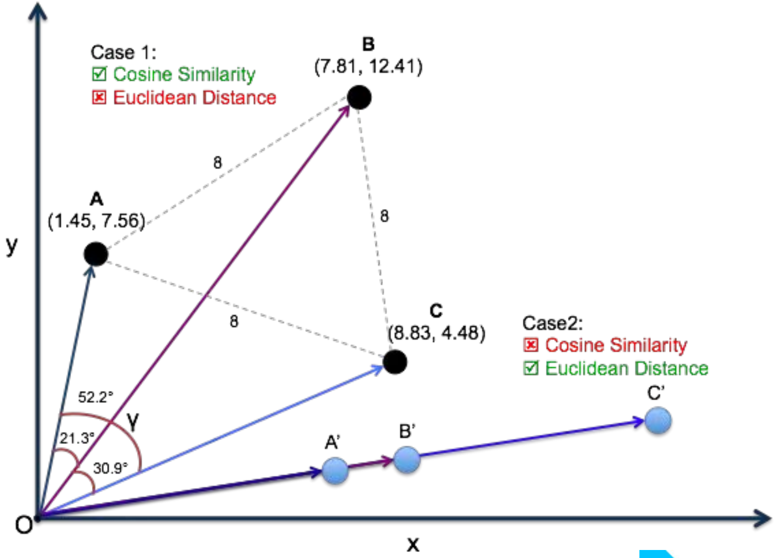
https://bit.ly/pinecone-icdar

infocusp
Innovations

Distance measures for retrieval

Distance measures used for retrieval

- In most cases, Euclidean (l2) distance or cosine similarity are used
- Cosine distance = 1- cosine similarity
- Euclidean is less expensive but allows limited space
- Conditions for distance metric 🙂



Source: https://medium.com/@sasi24/cosine-similarity-vs-euclidean-distance-e5d9a9375fc8

Approximate nearest neighbours

- For large number of embeddings stored in the vector DB, almost always approximate nearest neighbors is used compared to nearest neighbor
- Requires tuning of parameters for optimal speed/ accuracy
- Faster retrieval
- Locality sensitive hashing/ KD trees

infocusp Innovations

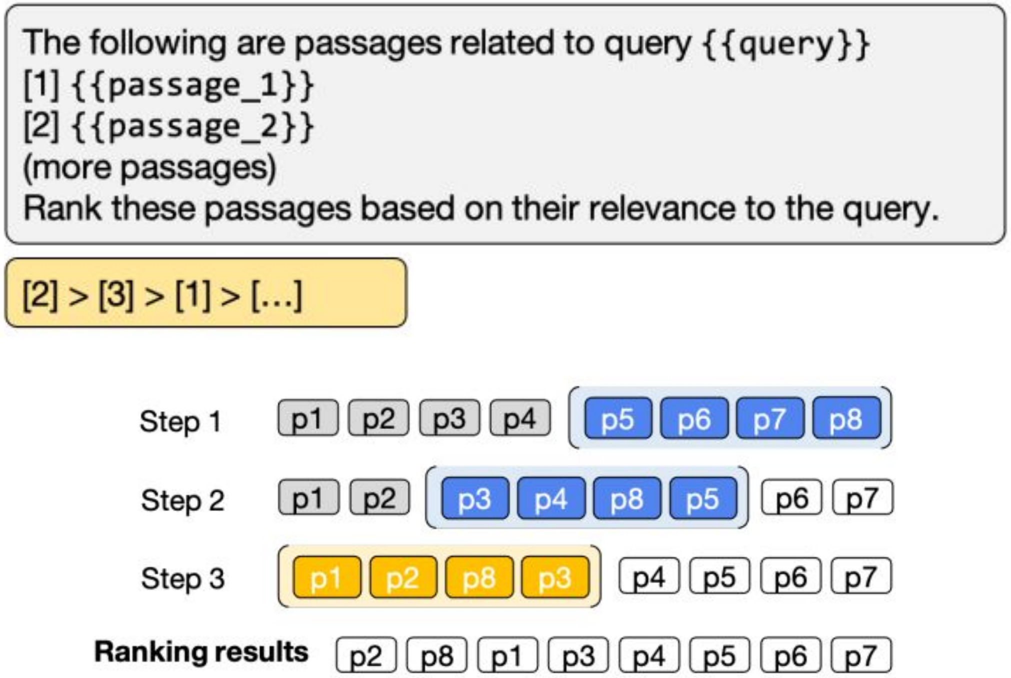
End to end RAG in python

https://bit.ly/e2e-langchain

Improving RAG: Reranking

Reranking of the retrieved responses

RankGPT (2023 EMNLP outstanding paper recipient) GitHub paper



infocusp Innovations

system:
You are RankGPT, an intelligent assistant that can rank passages based on their relevancy to the query.

user:
I will provide you with {{num}} passages, each indicated by number identifier []. Rank them based on their relevance to query: {{query}}.

assistant:
Okay, please provide the passages.

user:
[1] {{passage_1}}

assistant:
Received passage [1]

user:
[2] {{passage_2}}

assistant:
Received passage [2]

(more passages) ...

user
Search Query: {{query}}.
Rank the {{num}} passages above based on their relevance to the search query. The passages should be listed in descending order using identifiers, and the most relevant passages should be listed first, and the output format should be [] > [], e.g., [1] > [2]. Only response the ranking results, do not say any word or explain.

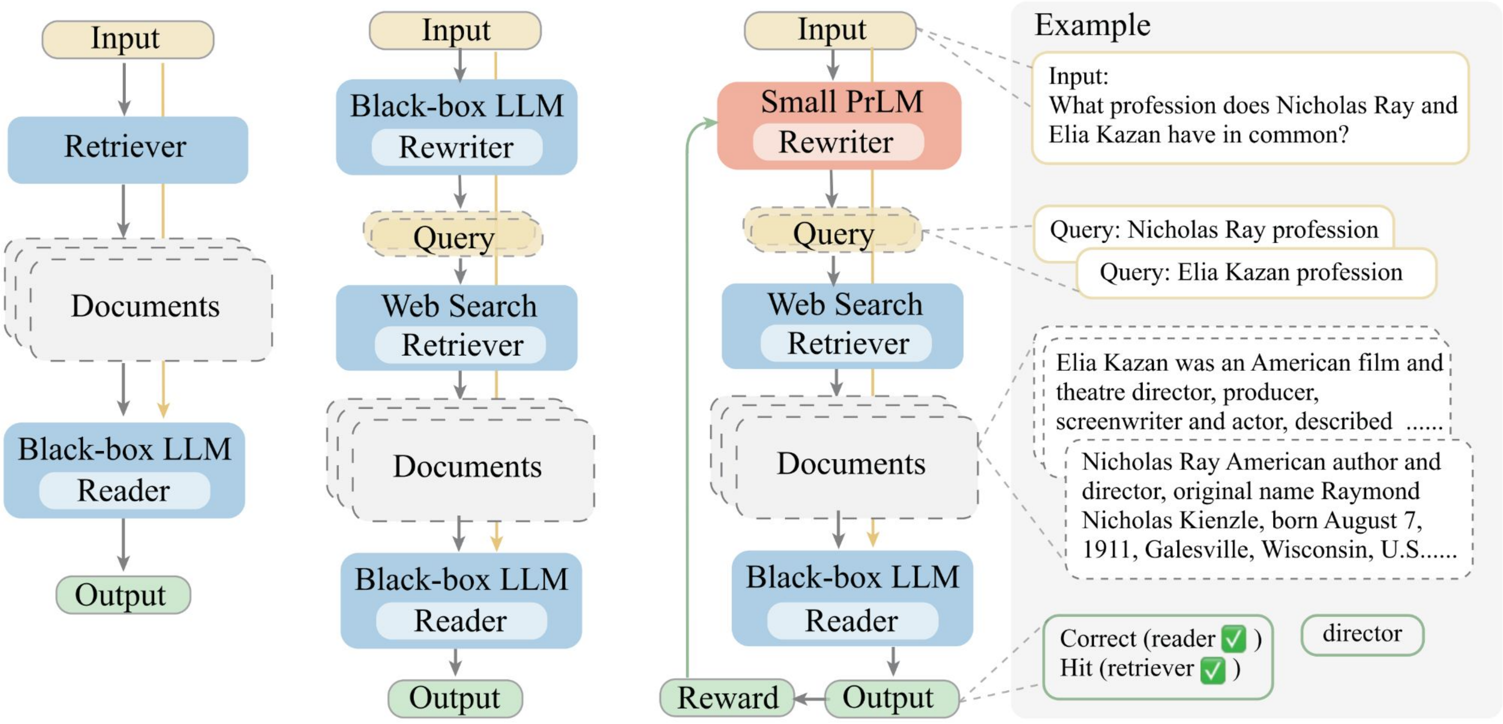
infocusp Innovations

Hands on reranking

https://bit.ly/reranking-icdar

Improving RAG: query rewriting

Query rewriting



(a) Retrieve-then-read (b)Rewrite-retrieve-read (c) Trainable rewrite-retrieve-read

T5-large as the rewriter, ChatGPT and Vicuna-13B as the LLM reader.

Query rewriting

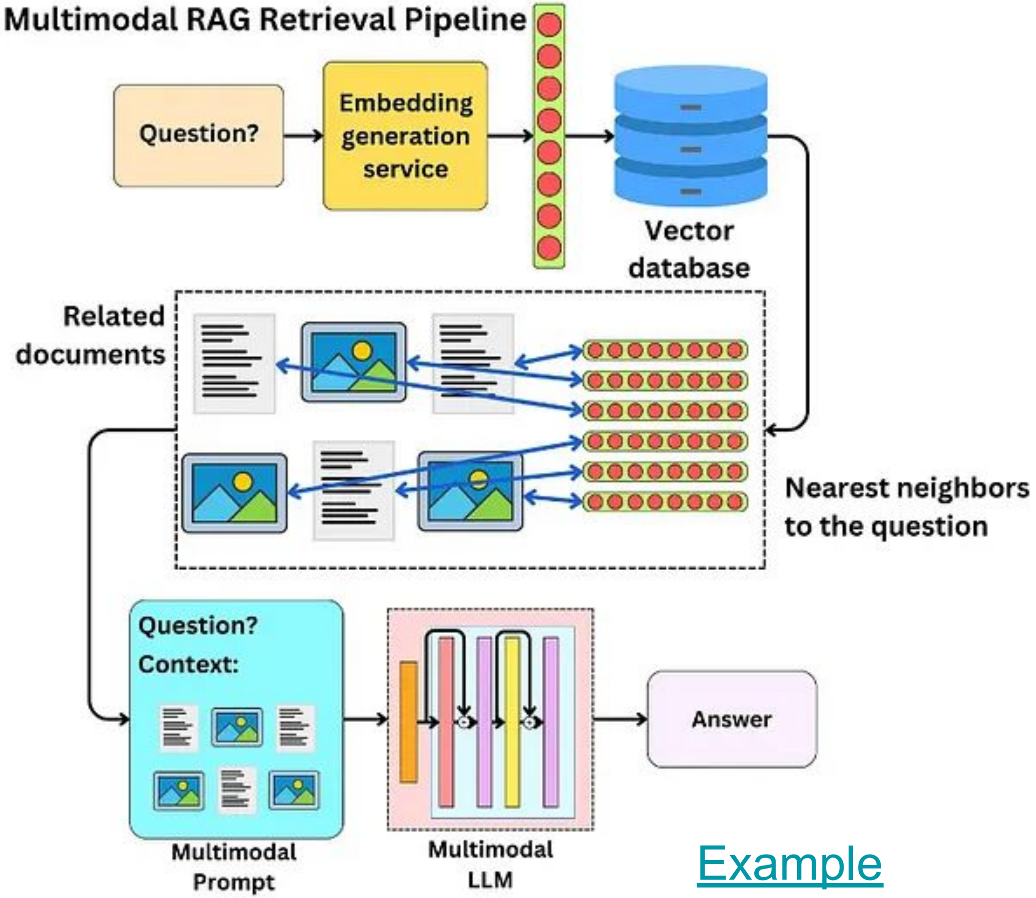
“What science fantasy young adult series, told in first person, has a set of companion books narrating the stories of enslaved worlds and alien species?"
"generated_text": "science fantasy young adult series; companion books narrating enslaved worlds and alien species",

“What is the name of the fight song of the university whose main campus is in Lawrence, Kansas and whose branch campuses are in the Kansas City metropolitan area?", "generated_text": "name of the university whose main campus is in Lawrence, Kansas; name of the university whose branch campuses are in the Kansas City metropolitan area; fight song of the university”

One more example: 🙂

[More examples]

Multi modal RAG

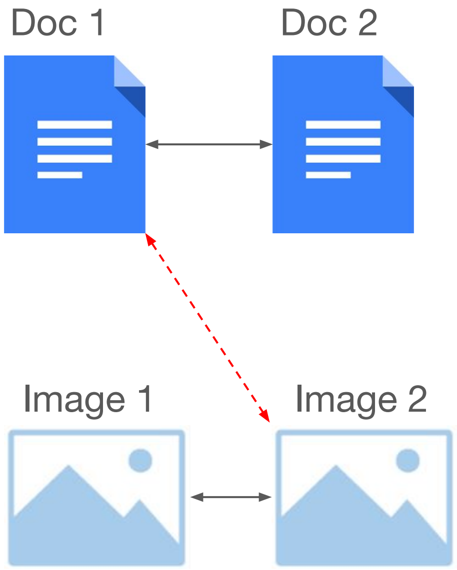


[Example]

What changed and which model could be used? 🙂

Source: https://medium.com/@bijit211987/multimodal-retrieval-augmented-generation-mm-rag-2e8f6dc59f11

Heart of multimodal RAG: CLIP



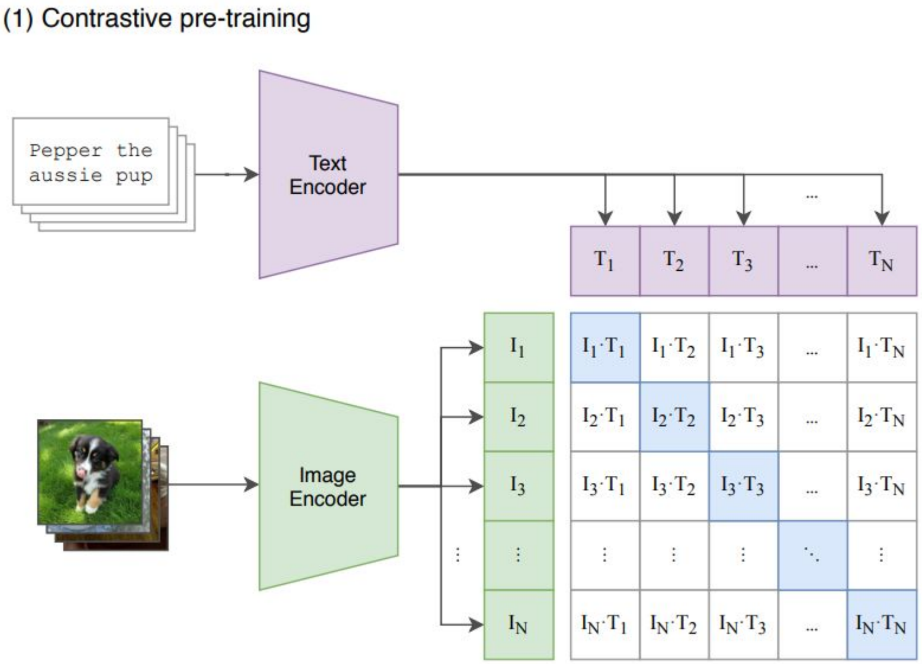
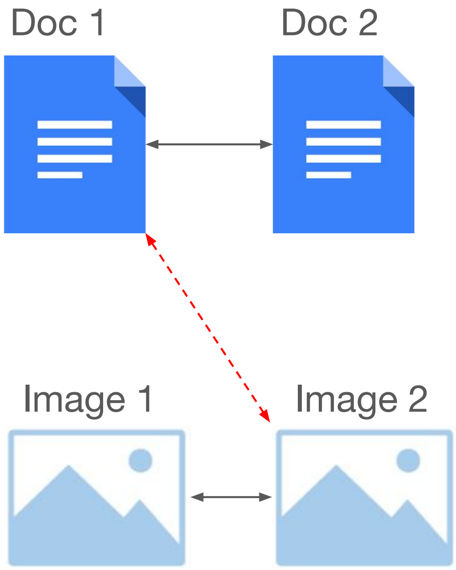
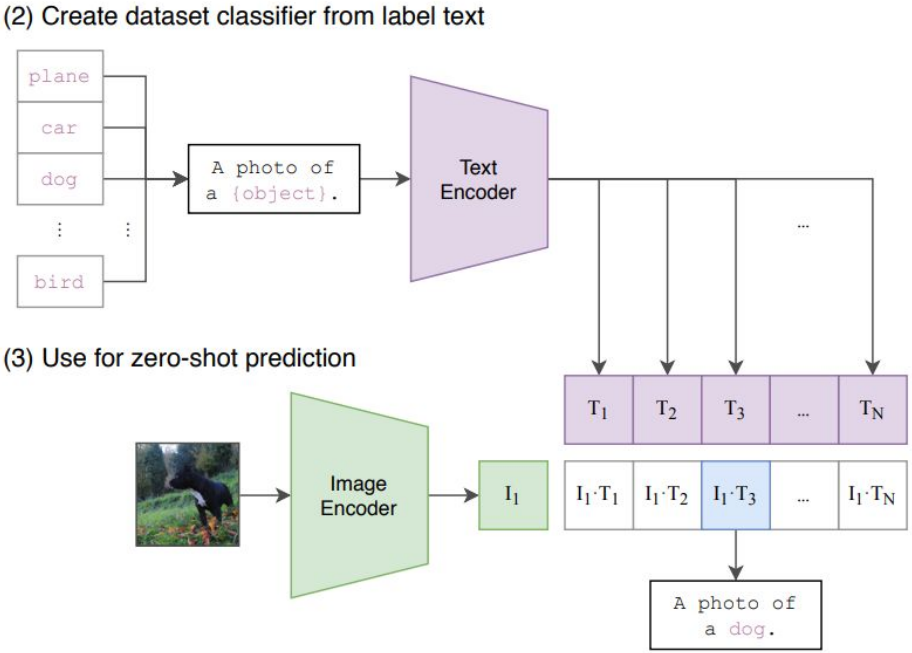


Image credits - [1]

[1] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.

Heart of multimodal RAG: CLIP





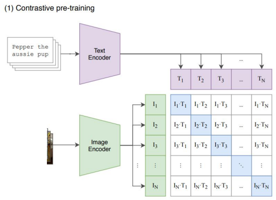


Image credits - [1]

[1] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.

Graph RAG

Graph RAG

- Uses LLM-generated knowledge graph to improve performance on complex Q&A
- This graph is used to perform prompt augmentation at query time
- Two main limitations of RAG it overcomes
    - Understanding complex disparate data
    - Understand semantic concepts over large data collections end to end (Query focused summarization)
- Example of both being overcome by Graph RAG

infocusp Innovations

GraphRAG Process

Index

- Chunk the input text into smaller chunks
- Extract all entities, relationships, and key claims from the chunks using an LLM.
- Incrementally group together
- Generate summaries of each community and its constituents from the bottom-up.

GraphRAG process

Query

- At query time, these structures are used to provide materials for the LLM context window when answering a question. The primary query modes are:
  - Global Search for reasoning about holistic questions about the corpus by leveraging the community summaries.
  - Local Search for reasoning about specific entities by fanning-out to their neighbors and associated concepts.

Datasets used by Graph RAG

- Podcast transcripts. Compiled transcripts of podcast conversations between Kevin Scott, Microsoft CTO, and other technology leaders (Behind the Tech, Scott, 2024). Size: 1669 × 600-token text chunks, with 100-token overlaps between chunks (~1 million tokens).
- News articles. Benchmark dataset comprising news articles published from September 2013 to December 2023 in a range of categories, including entertainment, business, sports, technology, health, and science

infocusp
Innovations

Evaluation of RAG systems

RAGAS

ragas score

generation

faithfulness
how factually acurate is the generated answer

answer relevancy
how relevant is the generated answer to the question

retrieval

context precision
the signal to noise ratio of retrieved context

context recall
can it retrieve all the relevant information required to answer the question

[Ragas library]()

Faithfulness

- Measures the factual consistency of the generated answer against the given context
- Range (0,1)

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

Answer Relevance

Focuses on assessing how pertinent the generated answer is to the given prompt.

The mean cosine similarity of the original question to a number of artificial questions, which were generated (reverse engineered) based on the answer

Context precision

Context Precision is a metric that evaluates whether all of the ground-truth relevant items present in the contexts are ranked higher or not.

Uses LLM to evaluate

Context recall

Context recall measures the extent to which the retrieved context aligns with the annotated answer, treated as the ground truth.

$$\text{context recall} = \frac{|\text{GT claims that can be attributed to context}|}{|\text{Number of claims in GT}|}$$

Thank you!

infocusp
Innovations

Keep in touch

falak@infocusp.com