UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
Dipartimento di
Informatica, Sistemistica
e Comunicazione

Introduction to RAG: Retrieval-Augmented Generation

Prof. Marco Viviani
Department of Informatics, Systems, and Communication (DISCo)

Background: IR + Text Generation

What is RAG?

- Integrating **Information Retrieval** (IR) Techniques in **Text Generation**

Information Retrieval

+

Text Generation

Close-book exam

=

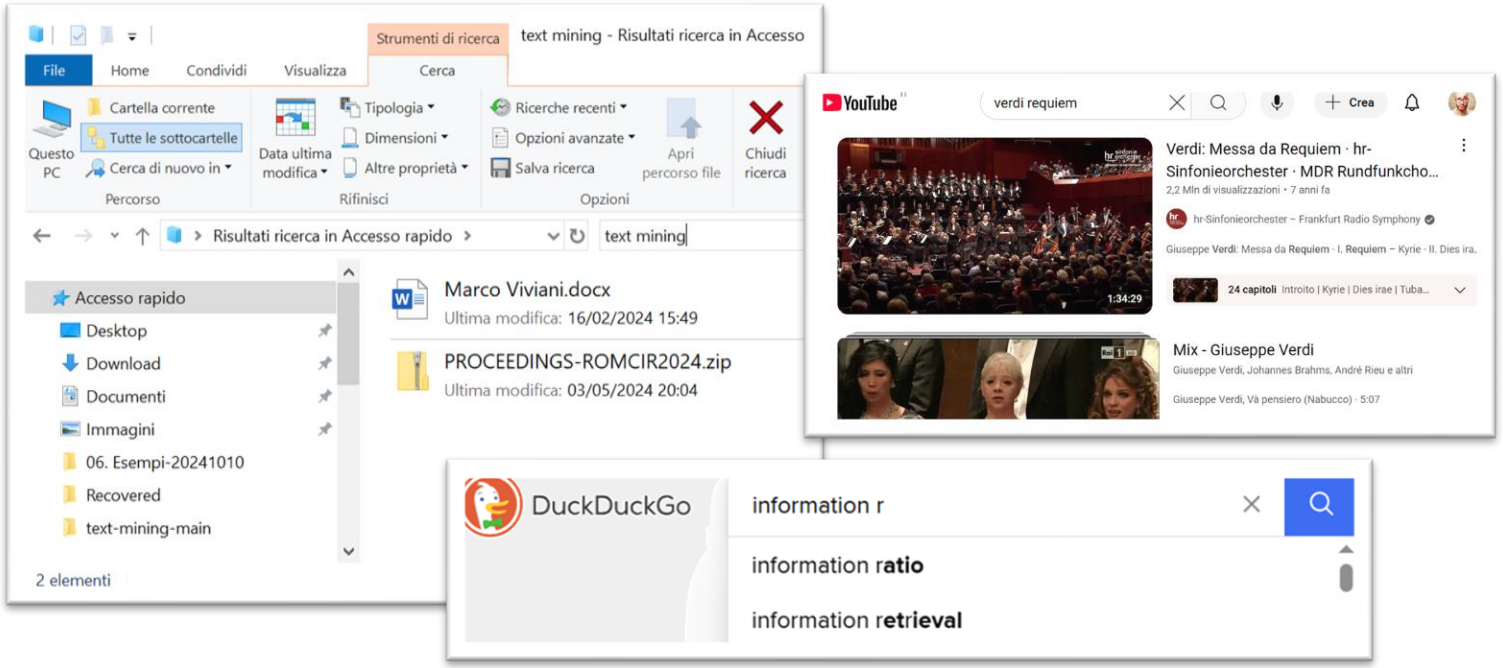
Retrieval-Augmented Text Generation

Open-book exam

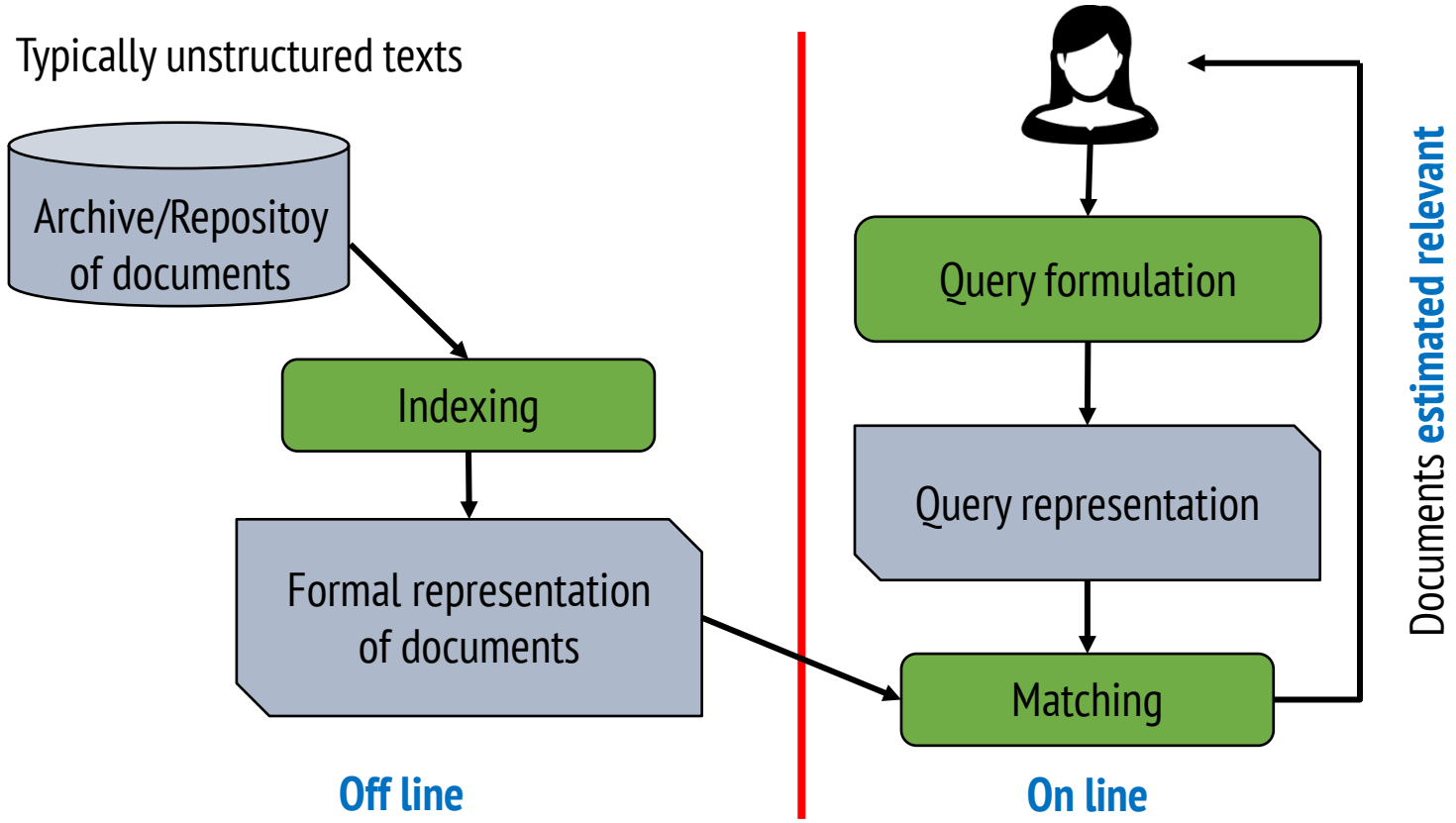
Information Retrieval

- **Information Retrieval** (IR) is the process of retrieving unstructured content (typically text) from large collections to satisfy a user's information need

- **Distinct forms** of IR
  - Desktop search
  - Web search
  - Vertical search
    - Video search
    - Audio search
    - …
  - …



Architecture of an IR System (Search Engine)



Information Retrieval Models

- An IRS is based on a **mathematical model** (IR model) that provides a formal description:
  - of the document
  - of the query
  - of how to compare the query and the document representations to estimate the relevance of documents to the query

- It should be noted that the use of the **same formal framework** to represent both documents and queries guarantees a correct matching

- IMPORTANT: An IR system simplifies the complexity of the retrieval activity → the results produced are not «perfect» (**estimate of relevance**)

The Notion of Relevance

- Retrieving the relevant documents for the user → The **relevance** of a document is relative to the formulated query (i.e., topical relevance, a.k.a. topicality)
  - Nowadays → Multi-dimensional relevance, i.e., topicality "+" novelty, popularity, factual accuracy, …

- **Exact comparison**: "binary" notion of relevance
  - Relevant / Not relevant

- **Partial comparison**: "gradual" notion of relevance:
  - Idea: comparison between the document and the query that tolerates mismatches (e.g., similarity of the query to the "document")

Sparse VS Dense Retrieval

Sparse Retrieval

- Represents queries/docs as sparse vectors (mostly zeros)
  - Bag-of-Words, TF-IDF
- Based on term matching (e.g., VSM, BM25)
- Relies on exact keyword overlaps
- Fast and interpretable
- Limited in capturing semantic meaning

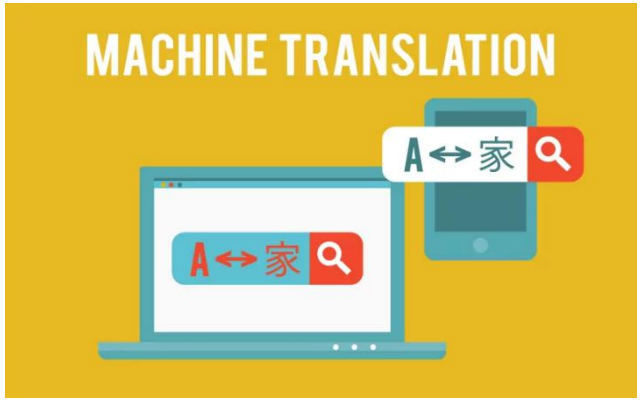
Dense Retrieval

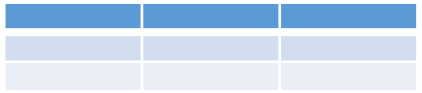
- Represents queries/docs as dense vectors
  - Neural embeddings (e.g., BERT-based)
- Captures semantic similarity, not just keywords
- Requires more compute (Approximate Nearest Neighbor search)
- Often more effective in open-domain QA and semantic search

Text Generation

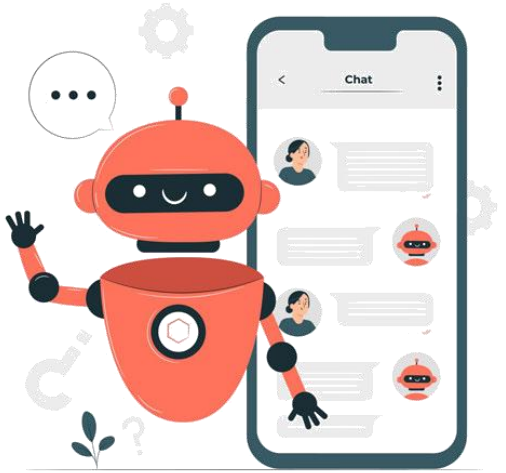
- **Text generation**, also known as **Natural Language Generation** (NLG), is the task of automatically producing coherent and contextually relevant text, to approximate or replicate human-written language

- Several applications
  - Machine translation
  - Open-ended text generation
  - Summarization
  - Dialogue generation / Chatbots
  - Data-to-text generation
  - …





Text text text text text text text …



NLG and Language Modeling

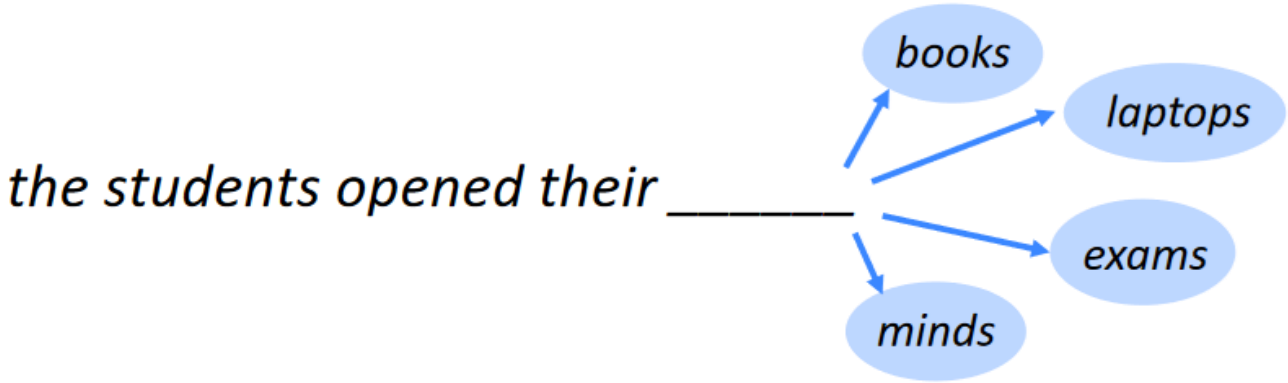
- Natural Language Generation (NLG) and **Language Modeling** (LM) are deeply connected – you can think of language modeling as the foundation of many NLG systems

- A language model is a **probabilistic model** that estimates the likelihood of a given sequence of words:
  - $P(w_1, w_2, \ldots, w_n)$
  - This is useful for evaluating how "likely" ("fluent") a sentence is
  - Example: Estimating the likelihood of "The cat sat on the mat" vs. "Cat the on sat mat the"

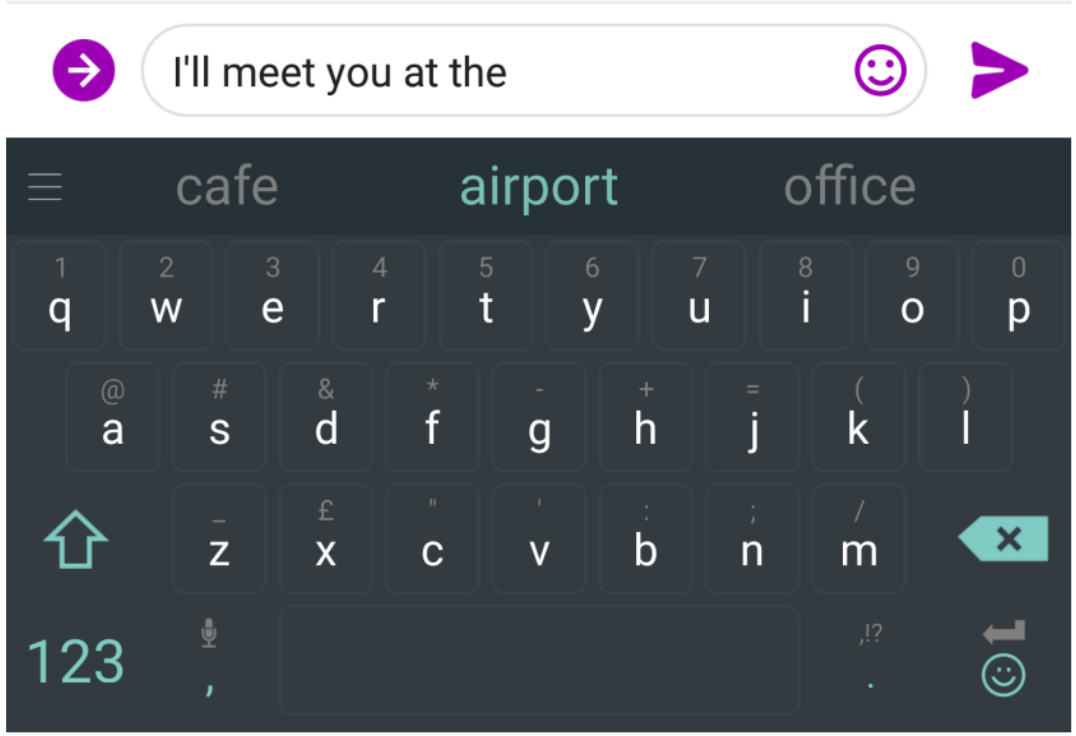
Predicting the Next Word

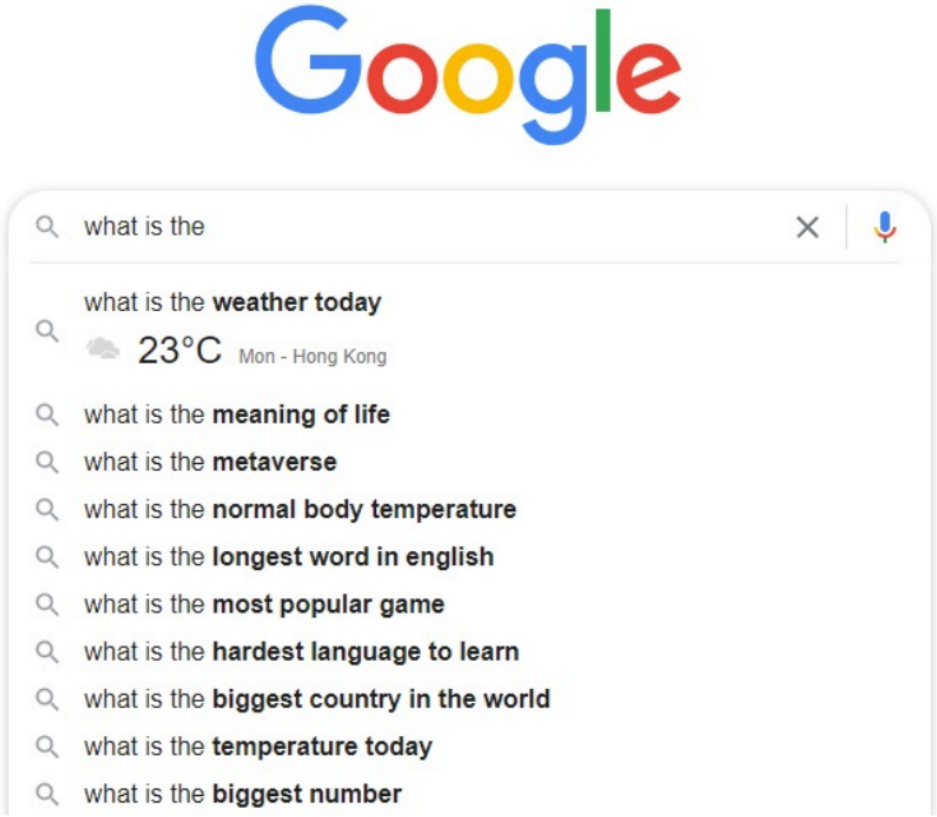
- It can be regarded as a probabilistic mechanism for **"generating" text**, thus also called a "generative" model

- The language model learns to predict the next word given the previous ones
  - $P(w_t \mid w_1, w_2, \ldots, w_{t-1})$



LMs in Everyday Life!





Traditional (Pre-Deep Learning) Way

n-gram LMs

- Collect statistics about how frequent different n-grams are (Auto-regressive LM / Causal Language Modeling)
- 2-gram (bigram) LM: the probability of a word in a sequence depends on the word that precedes it
- 3-gram (trigram) LM: the probability of a word in a sequence depends on the two words that precede it
- Example of a 4-gram LM (prediction based on the previous three words):
  - ~~As the proctor started the clock, the~~ students opened their ____

$$P(\boldsymbol{w}|\text{students opened their}) =$$

$$= \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

For example, suppose that in the corpus:

- "students opened their" occurred 1,000 times
- "students opened their books" occurred 400 times
  → P(books | students opened their) = 0.4
- "students opened their exams" occurred 100 times
  → P(exams | students opened their) = 0.1

Some Issues with n-gram LMs

- **Sparsity**
  - Hard to compute the probability of unseen text
- **Storage**
  - Need to store count for all n-grams. Increasing n or corpus increases model size!
- **Insufficient model of language**
  - Language has long-distance dependencies: "The computer which I had just put into the machine room on the fifth floor crashed"

Going beyond n-gram LMs

Neural Language Models

- Use neural networks to learn word representations and model longer dependencies
- Milestones:
  - Word2Vec / GloVe (Static embeddings)
  - RNNs / LSTMs (Model sequences better than n-grams)
- Limitations: Still struggles with long-term context, sequential computation is slow

Transformers and Pretrained LMs

- Attention-based architectures (e.g., Transformer by Vaswani et al., 2017)
  - Can process entire sequences in parallel (non-sequential)
- Pre-trained Language Models (PLMs):
  - BERT (2018): Masked Language Modeling (MLM) (bidirectional)
  - GPT (2018+): Causal Language Modeling (CLM) or Autoregressive Language Modeling (ALM) (left-to-right)

Large Language Models (LLMs)

- **Massive scale**: Billions of parameters, trained on diverse and massive corpora
- **Knowledge** is baked into weights
- **Self-supervised learning**. No labeled data needed

- **GPT (Generative Pre-trained Transformer)**:
  - GPT-1 (2018) → 117 million parameters, 985 million words
  - GPT-2 (2019) → 1.5 billion parameters
  - GPT-3 (2020) → 175 billion parameters. Chat GPT is also based on this model
  - GPT-4 (early 2023) → likely to contain trillions of parameters
  - GPT-4 Turbo (late 2023), optimized for efficiency → unspecified parameter count

https://www.geeksforgeeks.org/large-language-model-llm/

General-purpose, but not always task- or domain-specific

Optimizing LLMs

- **Fine-tuning** → Adapting a pre-trained model to a specific task or domain by training it further on a new, usually smaller, dataset
  - The model's weights are updated to perform better on the new task
    - Task-specific fine-tuning: Like text classification, question answering, or summarization
    - Domain-specific fine-tuning: Like medical, legal, or technical text
  - Costly, data-hungry, hard to update knowledge

- **In-context learning** → Teaching a language model to perform a task just by showing examples or instructions in the input **prompt**
  - No need to change model weights – just craft clever inputs ("prompts") to guide the model
  - Zero-/Few-shot learning → Generalize with minimal examples
    - Zero-shot: "Translate to French: 'Good morning'"
    - Few-shot: "Translate this to French: 'Good morning'. English = 'Good morning', 'French = 'Bonjour'"
  - Short-term → The model "learns" the task only during the current interaction
  - Not scalable for large systems

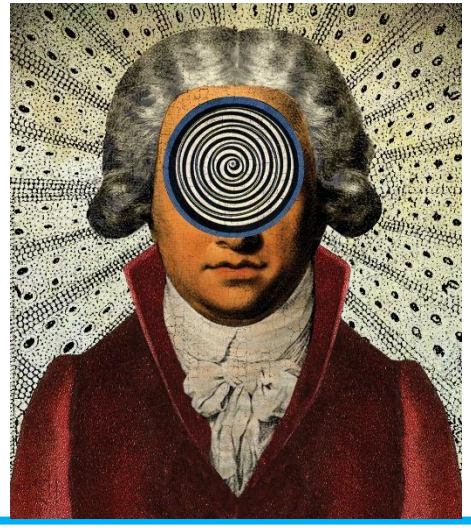
ALERT: LLMs Hallucinate!

- **Hallucination** refers to when the LLM generates information that is false, inaccurate, or made up, but it sounds convincing or plausible

Factual hallucination → "The capital of Italy is Milan."

Made-up data → "According to the latest data, 92% of people believe in extraterrestrial life"

(non-existing poll)



Fabricated citation → "According to a study by Smith et al. (2021), the human brain can process 10,000 thoughts per second"

(non-existing citation)

Why does Hallucination Happen? Solutions?

- **Contextual gaps** → The model may not have enough information to answer a query accurately and "fills the gap"
- **Overfitting to patterns** → The model learns patterns from large amounts of data, which could sometimes include incorrect information that gets reflected in its responses

- **Fine-tuning**
  - Fine-tune the model on a specific, curated dataset to reduce hallucination in that domain
- **Post-processing** and **Fact-checking**
  - Use of external fact-checking tools to verify outputs
- **Combine LLMs** with **External Knowledge** via **Retrieval-Augmented Generation (RAG)** → The core of this lecture

Retrieval-Augmented Generation

The Emergence of the Concept (2020)

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

https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Why RAG?

"LLMs' 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"

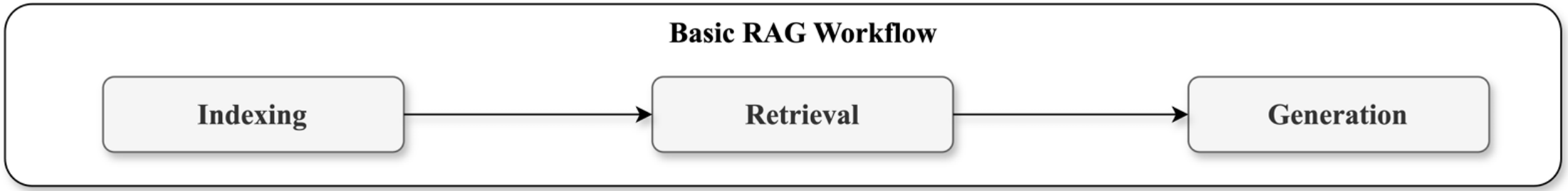
RAG: Basic Notions

- The idea behind RAG techniques is to **make use of knowledge "outside" the model** to provide a **"local" context (in-context)** that can supplement the model with appropriate knowledge without changing its parameters.

- These are basically prompting techniques that supplement the user's input with **contextual knowledge retrieved** by accessing external sources of information through a search engine.

Naive RAG

- During the nascent stages of RAG, its core framework is constituted by **indexing**, **retrieval**, and **generation**, a paradigm referred to as **Naive RAG**



Naive RAG: Indexing

- **Indexing** involves creating an inverted index—mapping each token to the documents/positions where it appears.
  - This stage involves text normalization processes such as tokenization, stemming, and the removal of stop words to enhance the text's suitability for indexing
- The integration of **Deep Learning** has revolutionized indexing through the use of pretrained LMs for generating semantic vector representations of texts
- When dealing with **Transformer models** and embedding-based search:
  - We often tokenize and chunk documents → Each chunk is a set of tokens that can fit within the model's context window
  - We generate embeddings for those chunks and index those

Naive RAG: Retrieval

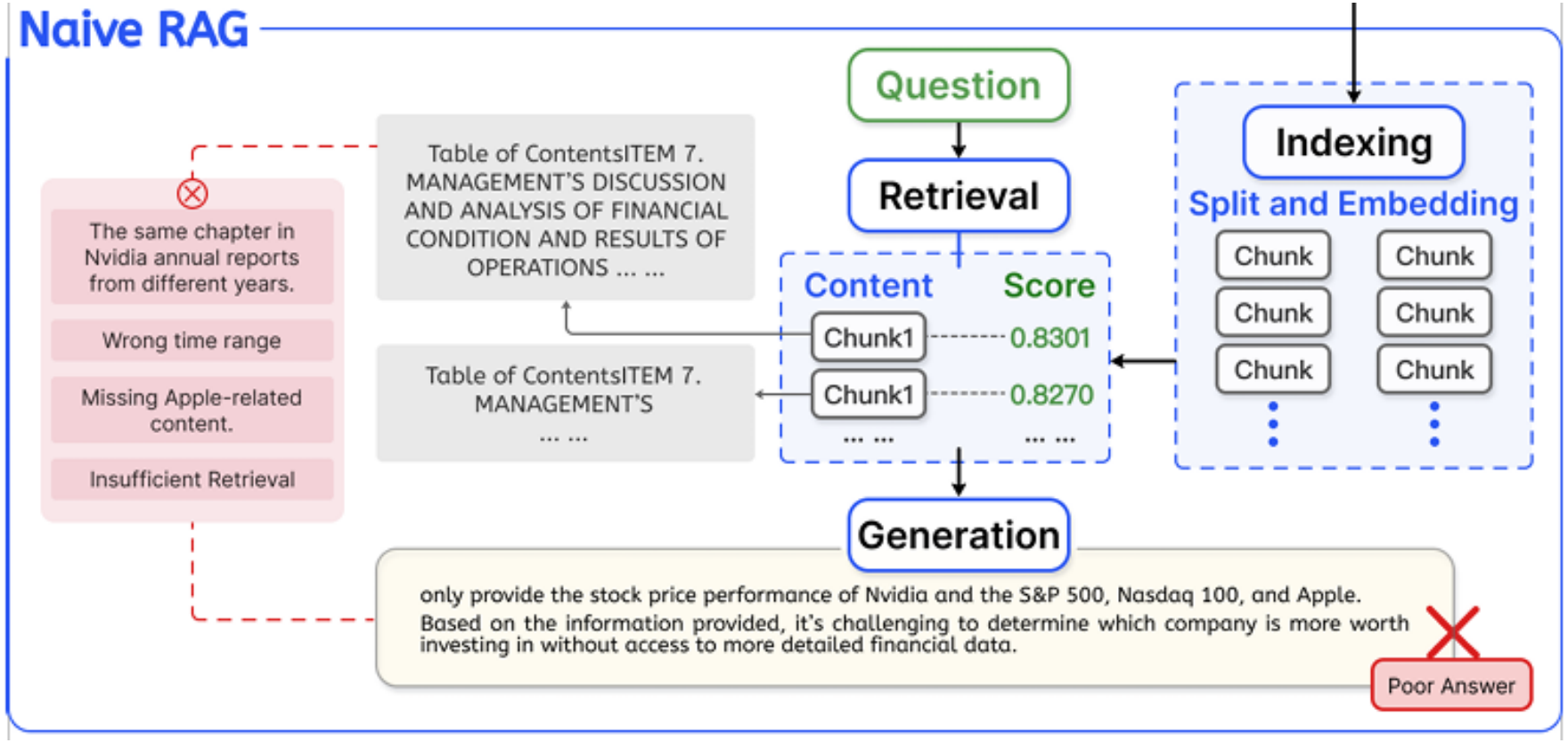
- **Traditional retrieval methods**, such as BM25, focus on term frequency and presence for document ranking → they often overlook the semantic information of queries

- **Current strategies** leverage pretrained LMs like BERT, which capture the semantic essence of queries more effectively → Dense retrieval models
  - They consider synonyms and the structure of phrases, thereby refining document ranking through the detection of semantic similarities
  - This is typically achieved by measuring vector distances between documents and queries, combining traditional retrieval metrics with semantic understanding to yield search results that are both relevant and aligned with user intent

Naive RAG: Generation

- The generation phase is tasked with producing text that is both **relevant to the query** and **reflective of the information** found in the retrieved documents
- The usual method involves **concatenating the query with the retrieved information**, which is then **fed into an LLM for text generation**
- The generated text should accurately convey the information from the retrieved documents and align with the query's intent, while also offering the flexibility to introduce new insights or perspectives not explicitly contained within the retrieved data

Naive RAG: An Example



Naive RAG: Issues

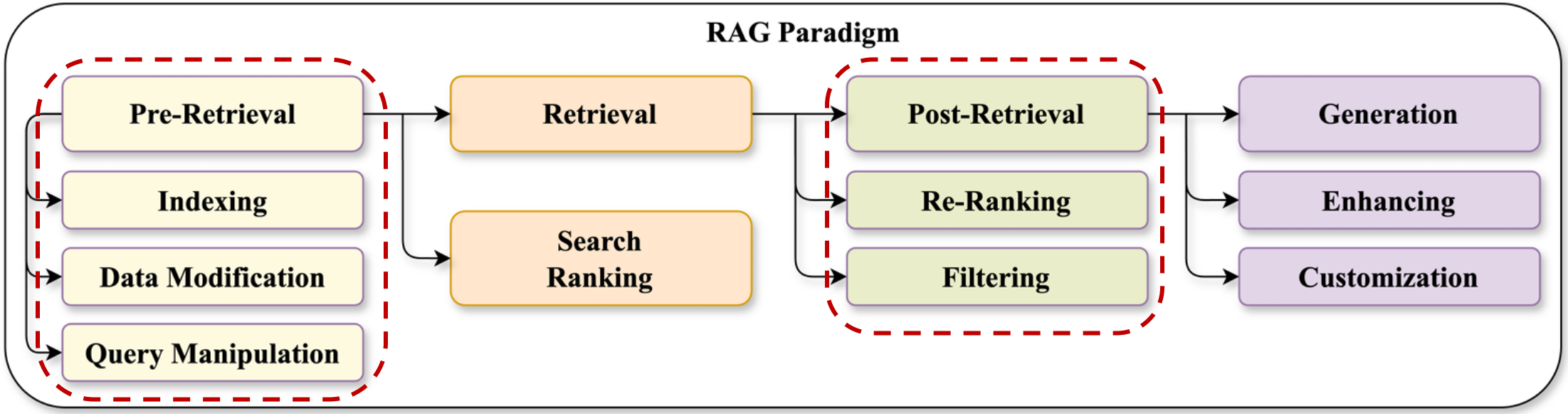
- **Shallow understanding of queries**
  - The semantic similarity between a query and a document chunk is not always highly consistent
  - Relying solely on similarity calculations for retrieval lacks an in-depth exploration of the relationship between the query and the document

- **Retrieval redundancy and noise**
  - Feeding all retrieved chunks directly into LLMs is not always beneficial
  - Research indicates that an excess of redundant and noisy information may interfere with the LLM's identification of key information, thereby increasing the risk of generating erroneous and hallucinated responses

Advanced RAG

- Advanced RAG focuses on **optimizing the retrieval phase**, aiming to enhance retrieval efficiency and strengthen the utilization of retrieved chunks
- Typical strategies involve **pre-retrieval processing** and **post-retrieval processing**
- For instance, query rewriting is used to make the queries clearer and more specific, thereby increasing the accuracy of retrieval, and the reranking of retrieval results is employed to enhance the LLM's ability to identify and utilize key information

Advanced RAG: Pipeline



Advanced RAG: Pre-Retrieval

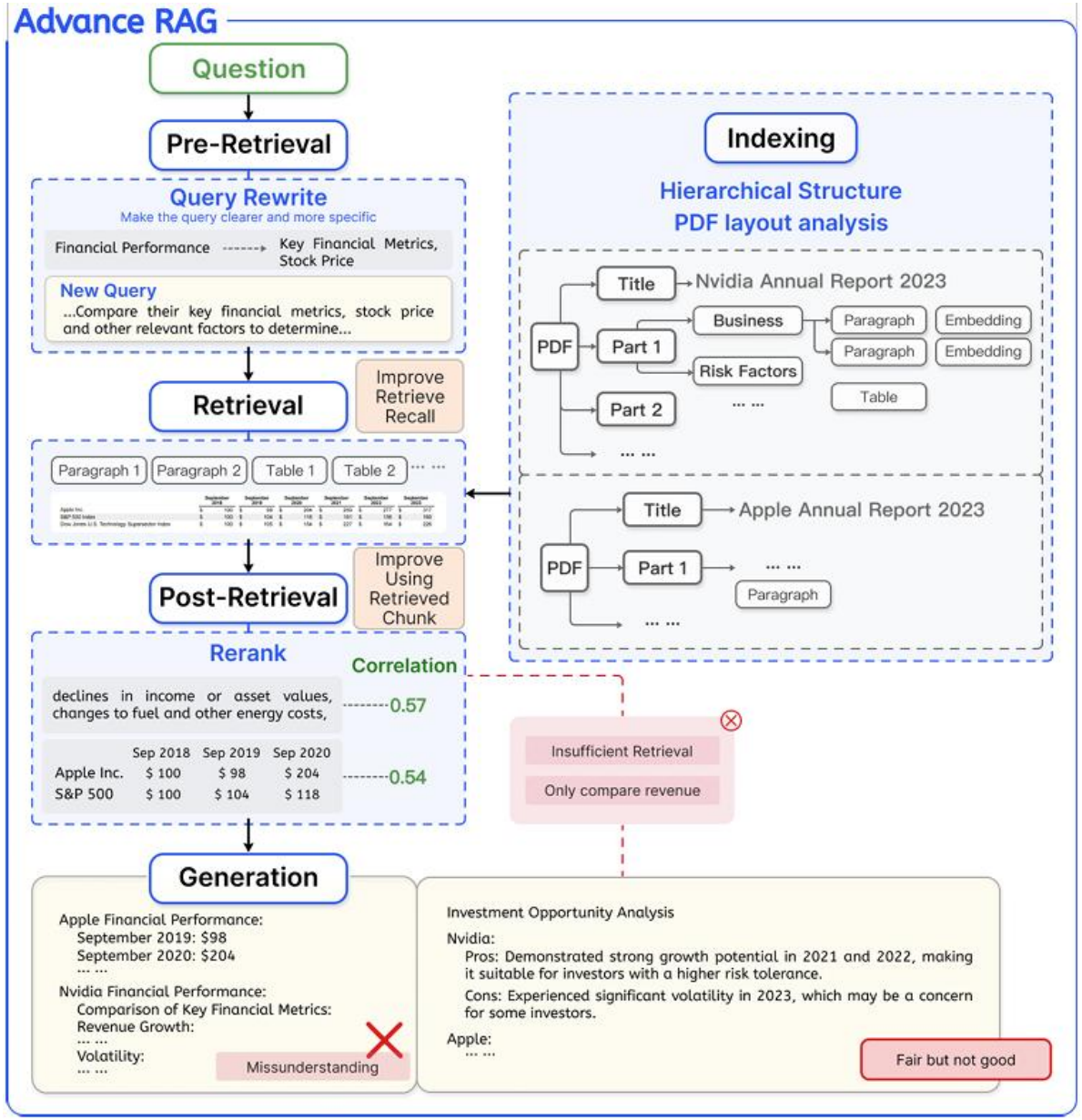
- The **specificity of indexing** depends on the task and data type
  - E.g., sentence-level indexing or paragraph-level indexing is better for Q-A systems to precisely locate answers, while document-level indexing is more appropriate for summarizing documents to understand their main concepts and ideas
- **Data modification** is also critical in enhancing retrieval efficiency
  - Preprocessing techniques → Removing irrelevant/redundant information and/or enriching the data with additional information, such as metadata, to boost the relevance and diversity of the retrieved content
- **Query manipulation** is performed to adjust user queries for a better match with indexed data
  - Query reformulation → Rewrites the query to align more closely with the user's intention
  - Query expansion → Extends the query to capture more relevant results through synonyms or related terms
  - Query normalization → Resolves differences in spelling or terminology for consistent query matching

Advanced RAG: Post-Retrieval

- In the **re-ranking** step, the documents previously retrieved are reassessed, scored, and reorganized
  - Objective → More accurately highlight the documents most relevant to the query and diminish the importance of the less relevant ones
  - Methods → Incorporating additional metrics and external knowledge sources to enhance precision

- **Filtering** aims to remove documents that fail to meet specified quality or relevance standards
  - Methods:
    - Establishing a minimum relevance score threshold to exclude documents below a certain relevance level
    - Using the feedback from users or prior relevance evaluations assists in adjusting the filtering process, guaranteeing that only the most relevant documents are retained for text generation

Advanced RAG: An Example



Advanced RAG: Issues

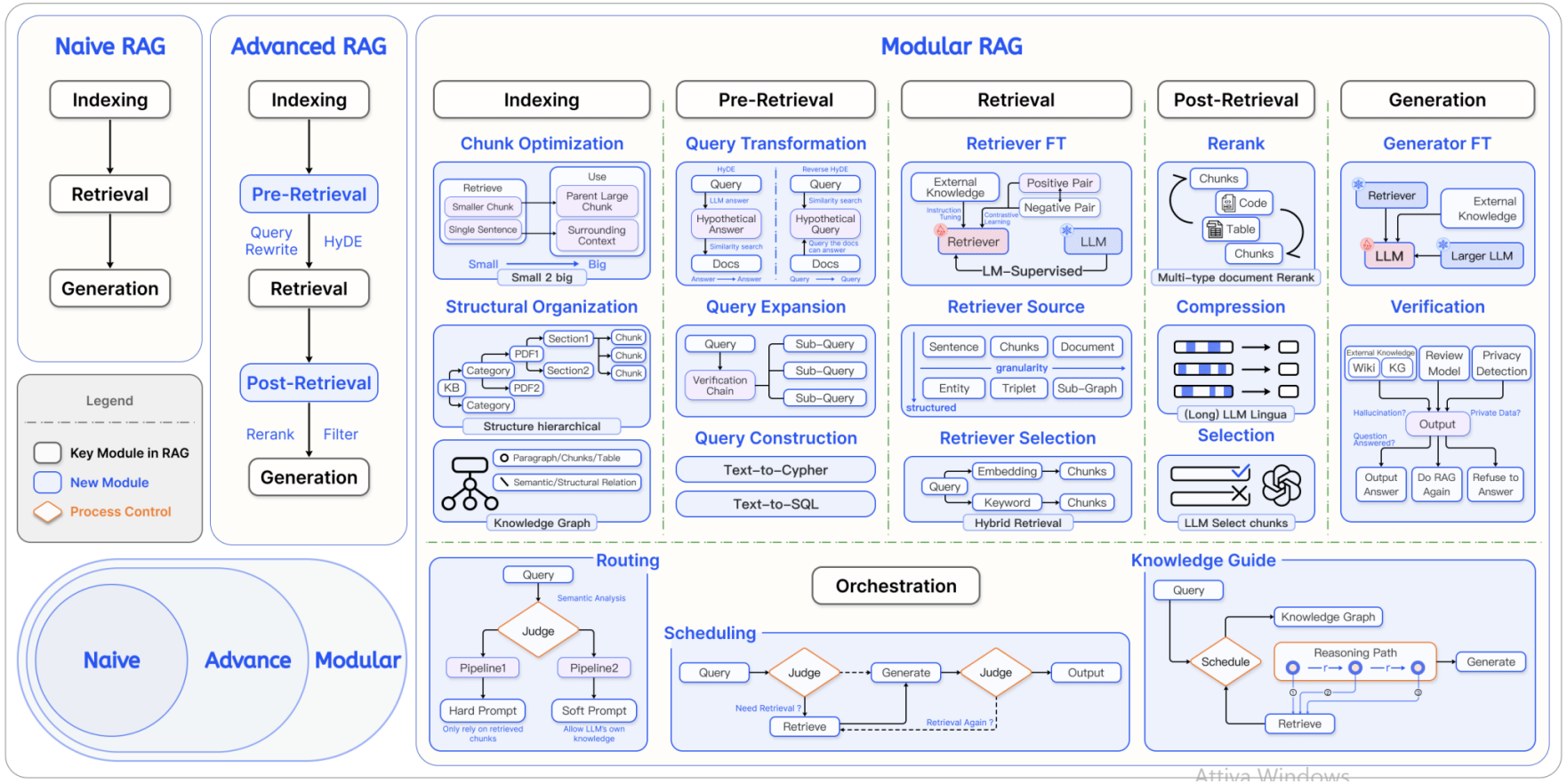
- There remains a **gap between RAG capabilities** and **real-world application requirements**

- RAG currently faces the following **new challenges**:
  - Complex data sources integration. RAGs are no longer confined to a single type of unstructured text data source but have expanded to include various data types (e.g., tables, knowledge graphs)
  - New demands for system interpretability, controllability, and maintainability
  - Component selection and optimization. More neural networks are involved in the RAG system, necessitating the selection of appropriate components to meet the needs of specific tasks and resource configurations
  - Workflow orchestration and scheduling. Components may need to be executed in a specific order, processed in parallel under certain conditions, or even judged by the LLM based on different outputs

Modular RAG

- The **current RAG paradigm** → Surpassing the traditional linear retrieval-generation paradigm
- **Modular RAG** → Consists of multiple independent yet tightly coordinated modules, each responsible for handling specific functions or tasks
- **Advantages of Modular RAG** → It enhances the flexibility and scalability of RAG systems
  - Users can flexibly combine different modules and operators according to the requirements of data sources and task scenarios

Modular RAG: Pipeline



Modular RAG: Some Optimization Modules

- **Indexing**
  - Chunk optimization
    - Sliding window
    - Metadata attachment
  - Structure organization
    - Hierarchical indexing
    - KG-based indexing
- **Pre-retrieval**
  - Query manipulation
    - LLM-based query expansion or rewriting
    - Multiple queries / sub-queries
- **Retrieval**
  - Sparse / Dense / Hybrid retrieval
  - Fine-tuning retrieval models
- **Post-retrieval**
  - Compress and select the retrieved content
  - Selection and removal of irrelevant chunks
- **Generation**
  - Generator fine-tuning
  - Verification (KB-verification, model-based verification)

Modular RAG: Orchestration

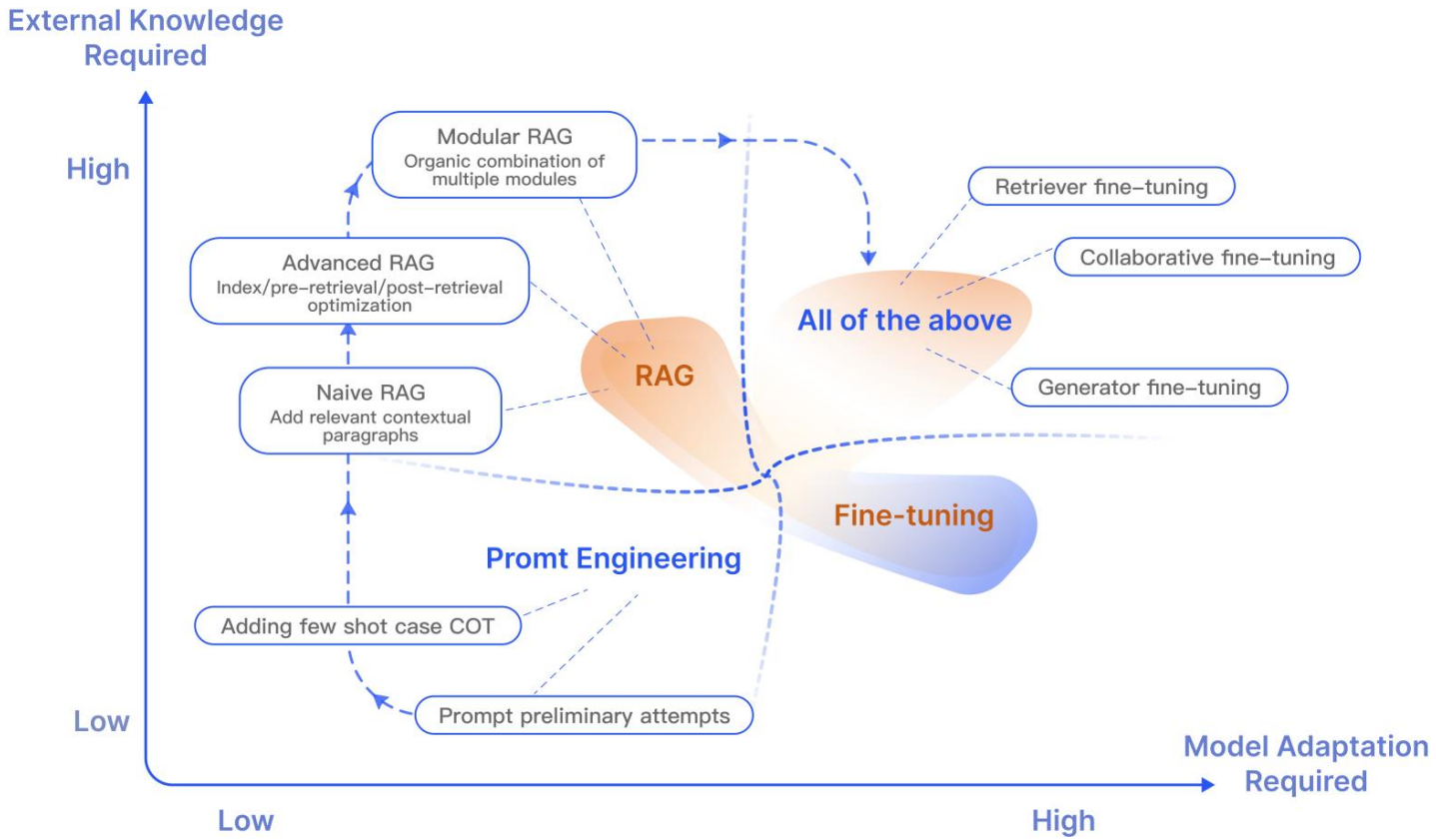
- Modular RAG incorporates **decision-making at pivotal junctures** and dynamically selects subsequent steps contingent upon the previous outcomes

- Routing
  - In response to diverse queries, the RAG system routes to specific pipelines tailored for different scenario, a feature essential for a versatile RAG architecture designed to handle a wide array of situations

- Fusion
  - Enhancing diversity by exploring multiple pipelines → Fusing for the best output

- Scheduling
  - It identifies critical junctures that require external data retrieval, assessing the adequacy of the responses, and deciding on the necessity for further investigation
  - It is commonly utilized in scenarios that involve recursive, iterative, and adaptive retrieval

RAG vs ALL


External Knowledge Required
High
Low
Low
High
Model Adaptation Required
Modular RAG
Organic combination of multiple modules
Advanced RAG
Index/pre-retrieval/post-retrieval optimization
Naive RAG
Add relevant contextual paragraphs
RAG
Retriever fine-tuning
Collaborative fine-tuning
All of the above
Generator fine-tuning
Fine-tuning
Promt Engineering
Adding few shot case COT
Prompt preliminary attempts

RAG: A Use Case Example

RAG and Health Information Retrieval

Discover Computing

Research

Enhancing Health Information Retrieval with RAG by prioritizing topical relevance and factual accuracy

Rishabh Upadhyay[1] · Marco Viviani[2]

Received: 23 August 2024 / Accepted: 17 February 2025
Published online: 01 April 2025

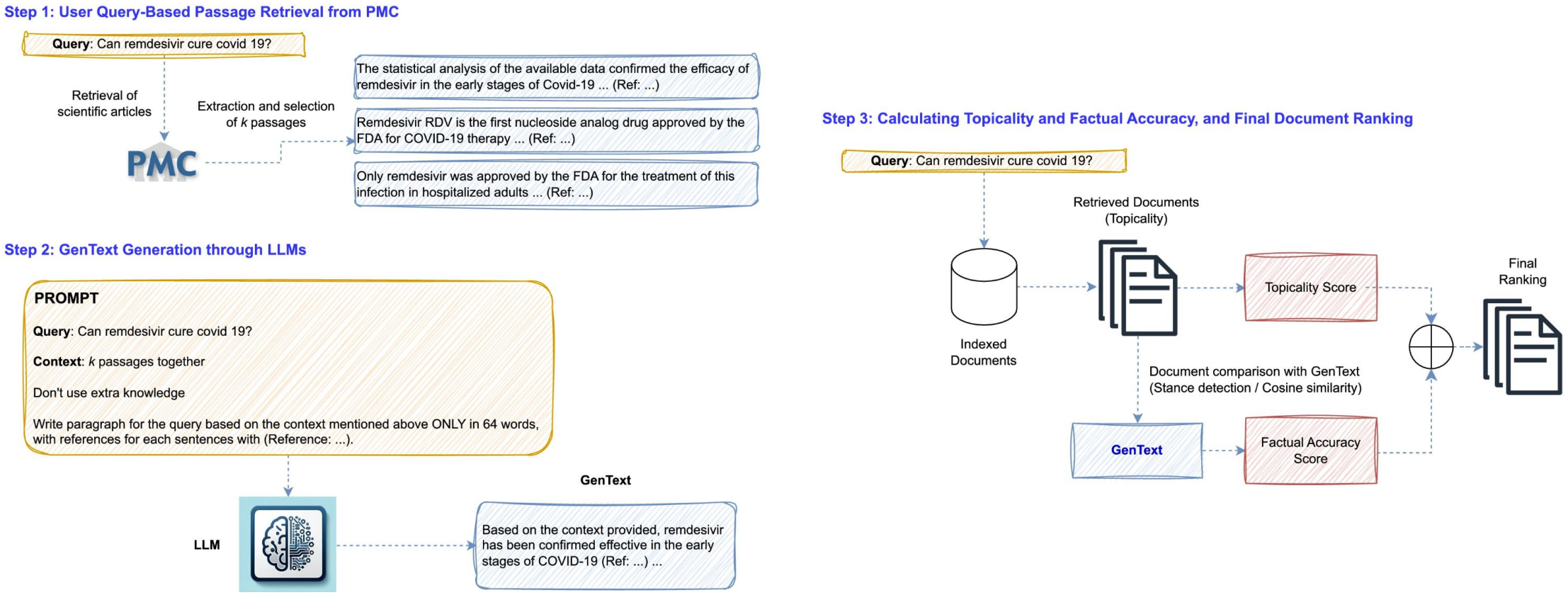
https://doi.org/10.1007/s10791-025-09505-5

The Proposed Solution

- **Integrating generative LLMs with a reputed, external knowledge**, such as the curated scientific repository of PubMed Central (PMC), a strategy designed to increase both the topical relevance and **factual accuracy** of the retrieved documents

- The proposed solution is characterized by **three key stages**:
  - User query-based passage retrieval from PMC
  - GenText generation through LLMs
  - Calculating topicality and factual accuracy, and final document ranking

The Proposed Solution: Pipeline



GenText generation through LLMs

LLM prompt

Query: can 5g antennas cause covid 19

Context: People around me told me not to get vaccinated against COVID-19 and reason 12 5G antennas are linked to the COVID-19 pandemic. At the same time there was no statistically significant difference in the average values of their answers regarding these reasons (Reference: 10316077). Interference can have a significant impact on 5G networks particularly in the context of Internet of Things IoT devices. (Reference: 10144169) These measures ensure that user privacy is protected and 5G networks can be trusted to handle massive data securely. The main causes and consequences of these challenges are summarized in Table 10 (Reference: 10255561). The need to deal with the explosion of multimedia services has been considered in the 6G network which will provide greater QoS while also guaranteeing QoE (Reference: 10347022). The importance of this was well proven in pandemic conditions of Covid-19 2729 So that in most organizations employees used different communication networks to do their work and after that the scope of communication networks in organizations has always grown. (Reference: 10399785) Newly emerging variants of SARS-CoV-2 continue to pose a significant threat to global public health by causing COVID-19 epidemics (Reference: 10288941). 4 GHz transmit signal filters and amplifies the received signal and downconverts it to IF and then digitizes the signal according to the programmed parameter settings on the laptop (Reference: 9953371). These tags incorporate antennas that can collect power efficiency regarding radio frequency queries from the RFID transceiver. Software is a good platform and crosses successful evolution for the sensor in Nanotechnology and bio-industries. In the medical field the Healthcare system is usually used to monitor the condition of patients (Reference: 10258751). The controller of the level crossing must be equipped with antennas able to receive signals from the sensors in the train in a safe distance in the paper we suggested possible technologies of communication. The real-world application can be a mixed solution (Reference: 10384084). The presence of various manufacturers and the diverse applications of sensors in disaster scenarios contribute to the heterogeneity of these sensors hence hindering the integration and sharing of information 107108. Some disasters may cause sequent disasters. For example, seismic activity or inundations can cause floods (Reference: 10490738).

Write a paragraph answering the query based on the context provided above constituted by ONLY 64 words, with references for each sentence with (Reference:...).

Do not use extra knowledge.

Based on the context provided, there is a misconception linking 5G antennas to the 19 pandemic (Reference: 10316077). connection has no statistically significant to support it (Reference: 10316077). Instead, important to note that 5G networks play a role in ensuring secure data handling and enhancing user privacy (Reference: Moreover, SARS-CoV-2 variants remain the cause of COVID-19 outbreaks (Reference: 10288941).

Topicality, Factual Accuracy, and Final Ranking

- The **topicality score** T is computed for each document d in the document collection, and it is derived using the BM25 retrieval model against the considered query q

$$T(d, q) = \text{BM25}(d, q)$$

- The **factual accuracy score** F integrates two distinct metrics to assess the adherence of documents to be retrieved to scientific evidence within GenText G

$$F(d, G) = \alpha \cdot stance(d, G) + (1 - \alpha) \cdot \cos(d, G)$$

- The **final document ranking** is obtained by performing a linear combination of topicality and factual accuracy scores in order to obtain the Retrieval Status Value (RSV)

$$RSV(d, q, G) = \beta \cdot T(d, q) + (1 - \beta) \cdot F(d, g)$$

Some Results

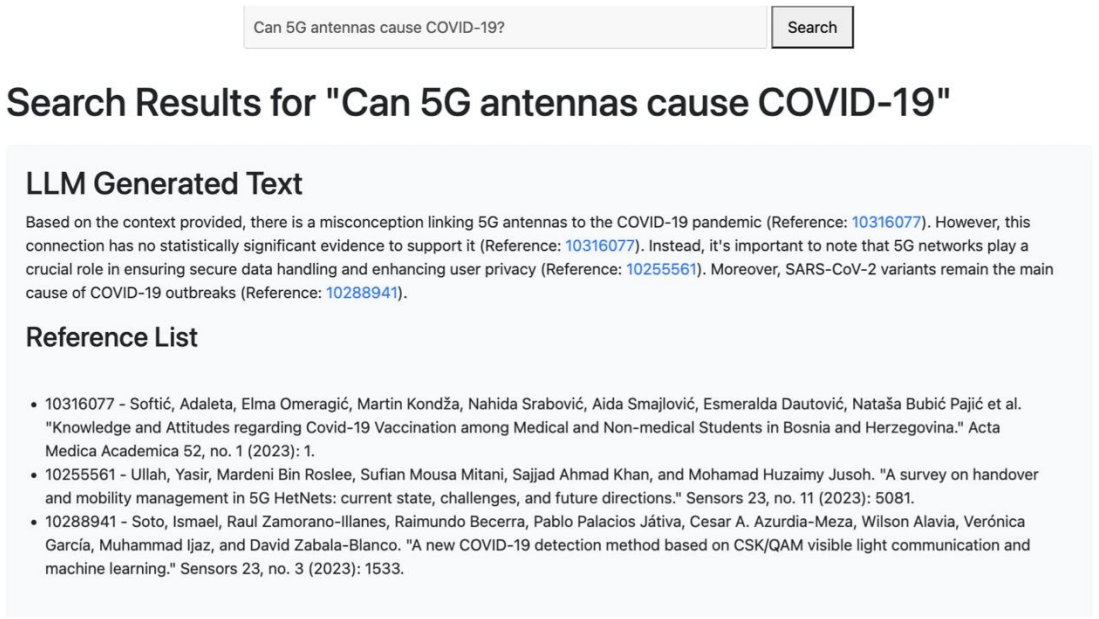
CLEF eHealth 2020 dataset

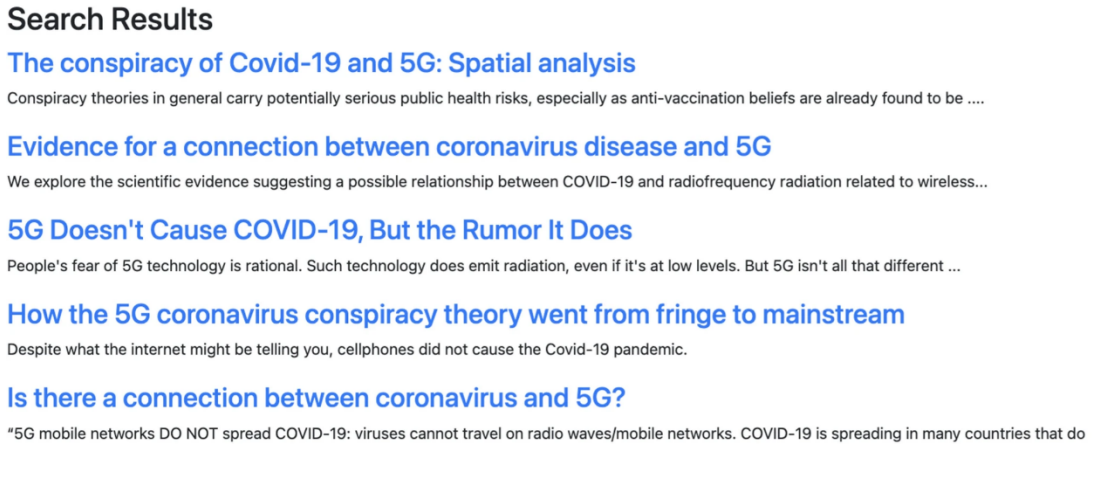
Model	CAM_{MAP}	CAM_{NDCG}	Embeddings
Top-5 Documents			
BM25	0.0431	0.1045	-
DigiLab	0.0433	0.1109	-
CiTIUS	0.0455	0.1119	-
WISE	0.0611	0.1198	BioBERT
$WISE_{NLI}$	0.0883	0.1823	BioBERT
GPT_{RAG}	0.1045	0.2098	BioBERT
$Llama_{RAG}$	**0.1079**	**0.2146**	BioBERT
$Falcon_{RAG}$	0.0994	0.2011	BioBERT
Top-10 Documents			
BM25	0.0784	0.1923	-
DigiLab	0.0823	0.1992	-
CiTIUS	0.0843	0.1999	-
WISE	0.1102	0.211	BioBERT
$WISE_{NLI}$	0.1302	0.2321	BioBERT
GPT_{RAG}	0.1502	0.2655	BioBERT
$Llama_{RAG}$	**0.1532**	**0.2702**	BioBERT
$Falcon_{RAG}$	0.1495	0.2568	BioBERT

TREC HM 2020 dataset

Model	CAM_{MAP}	CAM_{NDCG}	Embeddings
Top-5 Documents			
BM25	0.0631	0.1435	-
DigiLab	0.0712	0.1543	-
CiTIUS	0.0754	0.1554	-
WISE	0.0844	0.1608	BioBERT
$WISE_{NLI}$	0.0923	0.1922	BioBERT
GPT_{RAG}	0.1178	0.2234	BioBERT
$Llama_{RAG}$	**0.1222**	**0.2298**	BioBERT
$Falcon_{RAG}$	0.1123	0.2165	BioBERT
Top-10 Documents			
BM25	0.1047	0.2052	-
DigiLab	0.1186	0.2011	-
CiTIUS	0.1194	0.2095	-
WISE	0.1233	0.22	BioBERT
$WISE_{NLI}$	0.1341	0.2455	BioBERT
GPT_{RAG}	0.1547	0.2712	BioBERT
$Llama_{RAG}$	**0.1602**	**0.2723**	BioBERT
$Falcon_{RAG}$	0.1501	0.2665	BioBERT

A Tool for Explainability?





Some Bibliography

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in neural information processing systems, 33, 9459-9474.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2.
- Gao, Y., Xiong, Y., Wang, M., & Wang, H. (2024). Modular RAG: Transforming rag systems into lego-like reconfigurable frameworks. arXiv preprint arXiv:2407.21059.
- Huang, Y., & Huang, J. (2024). A survey on retrieval-augmented text generation for large language models. arXiv preprint arXiv:2404.10981.
- Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., ... & Li, Q. (2024, August). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 6491-6501).
- Upadhyay, R., & Viviani, M. (2025). Enhancing Health Information Retrieval with RAG by prioritizing topical relevance and factual accuracy. Discover Computing, 28(1), 27.