RAG with transformers

Outline

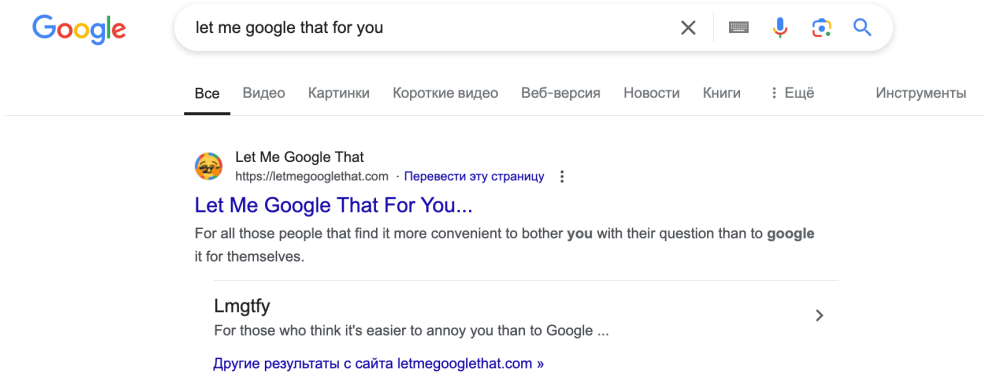
- Retriever-Augmented Generation (RAG)
- Retrieval
- Retrieval-Augmented Architectures
- RAG evaluation
- Intro to AI Agents
- AI Agents evaluation

Retriever-Augmented Generation

Motivation

Problem: LLMs are prone to lying and hallucinating.

Natural idea: give it ability to ”Google” factual information.



Solution: Retrieval-Augmented Generation (RAG).

Core idea: give LLM an access to some information database.

Why is RAG needed?

LLM challenges:

- May create “hallucinations” that are misleading.
- Rely on potentially outdated information.
- Ineffective when dealing with specific knowledge.
- Lack depth in specialized areas.
- Underdeveloped in the area of logical ability.
- Lack controllability.
- Cannot trace the source of knowledge.
- Cannot ensure confidentiality of data.
- The cost of their training is high.

How does RAG help?

- Improves **accuracy and relevance** using contextually significant knowledge from external sources.

RAG ≠ ”Search”

Искать в сети

🔍 **What does search do?**

- It performs a **real online search** (e.g., via Bing or another search engine).
- It returns **links and text snippets** from websites.
  The LLM then “**reads**” these pages and **uses the information** to generate an answer.

This is closer to **tool-augmented generation** or a **web-enhanced LLM**, but **not RAG** in the classical sense.

🧠 **What does RAG do?**

- It performs **retrieval** from a **local document store**, most often using **vector search**.
- Then it performs **generation** based on the **retrieved documents**.

All of this happens **inside the system**, **without any internet access**

RAG ≠ ”Search”, but…

Искать в сети

Sometimes, for clarity, we’ll illustrate certain search-related ideas using **search** as an example.

But remember: **Search and RAG are not exactly the same**.

These methods simply share a **common idea of accessing external knowledge**.

How does RAG help?

Question: *«У какой планеты в солнечной системе больше всего спутников?»* (ChatGPT vs Wikipedia)

ChatGPT without Search

На апрель 2025 года у **Сатурна** больше всего спутников в Солнечной системе!
У него официально подтверждено **145 спутников**, что делает его абсолютным рекордсменом.

How does RAG help?

Question: *«У какой планеты в солнечной системе больше всего спутников?»* (ChatGPT vs Wikipedia)

ChatGPT with Search

На апрель 2025 года **Сатурн** является планетой с наибольшим количеством спутников в Солнечной системе — у него подтверждено **274 спутника** . Это число значительно превышает количество спутников у Юпитера, у которого подтверждено 95 спутников .

В марте 2025 года астрономы объявили об открытии **128 новых спутников** Сатурна, что стало результатом многолетних наблюдений с использованием одного из самых мощных телескопов на Земле . Большинство этих новых спутников являются **неправильными** (irregular) — они имеют небольшие размеры (всего несколько километров в диаметре) и хаотичные орбиты, что свидетельствует о возможных столкновениях в прошлом .

skyatnightmagazine.com Guardian

How does RAG help?

Question: *«У какой планеты в солнечной системе больше всего спутников?»* (ChatGPT vs Wikipedia)

ChatGPT with Search

На апрель 2025 года **Сатурн** является планетой с наибольшим количеством спутников в Солнечной системе — у него подтверждено **274 спутника** . Это число значительно превышает количество спутников у Юпитера, у которого подтверждено 95 спутников .

В марте 2025 года астрономы объявили об открытии **128 новых спутников** Сатурна, что стало результатом многолетних наблюдений с использованием одного из самых мощных телескопов на Земле . Большинство этих новых спутников являются **неправильными** (irregular) — они имеют небольшие размеры (всего несколько километров в диаметре) и хаотичные орбиты, что свидетельствует о возможных столкновениях в прошлом .

skyatnightmagazine.com Guardian

Сатурн У **Сатурна** на март **2025** года известно 274 **спутника**, в том числе открытые до полёта «Вояджеров»: Титан (1655)

What is Retrieval-Augmented Generation (RAG)?

Retrieval-Augmented Generation (RAG) is:

- A method for **optimizing responses** generated by large language models (LLMs).
- It **combines text generation with information retrieval** from **a knowledge base**.
- It creates **more informative** and **accurate responses.**

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.

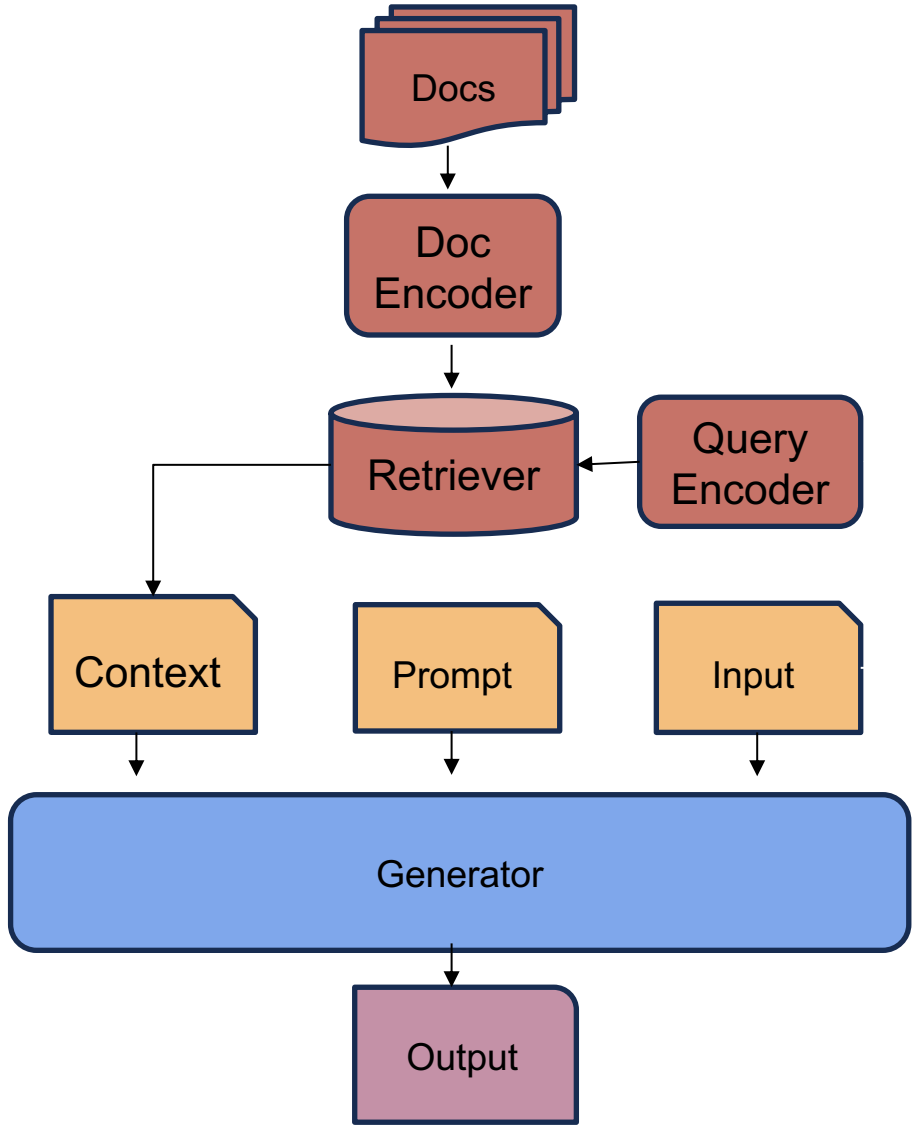
1 Introduction

Pre-trained neural language models have been shown to learn a substantial amount of in-depth knowledge from data [47]. They can do so without any access to an external memory, as a parameterized implicit knowledge base [51, 52]. While this development is exciting, such models do have downsides: They cannot easily expand or revise their memory, can't straightforwardly provide insight into their predictions, and may produce "hallucinations" [38]. Hybrid models that combine parametric memory with non-parametric (i.e., retrieval-based) memories [20, 26, 48] can address some of these issues because knowledge can be directly revised and expanded, and accessed knowledge can be inspected and interpreted. REALM [20] and ORQA [31], two recently introduced models that combine masked language models [8] with a differentiable retriever, have shown promising results,

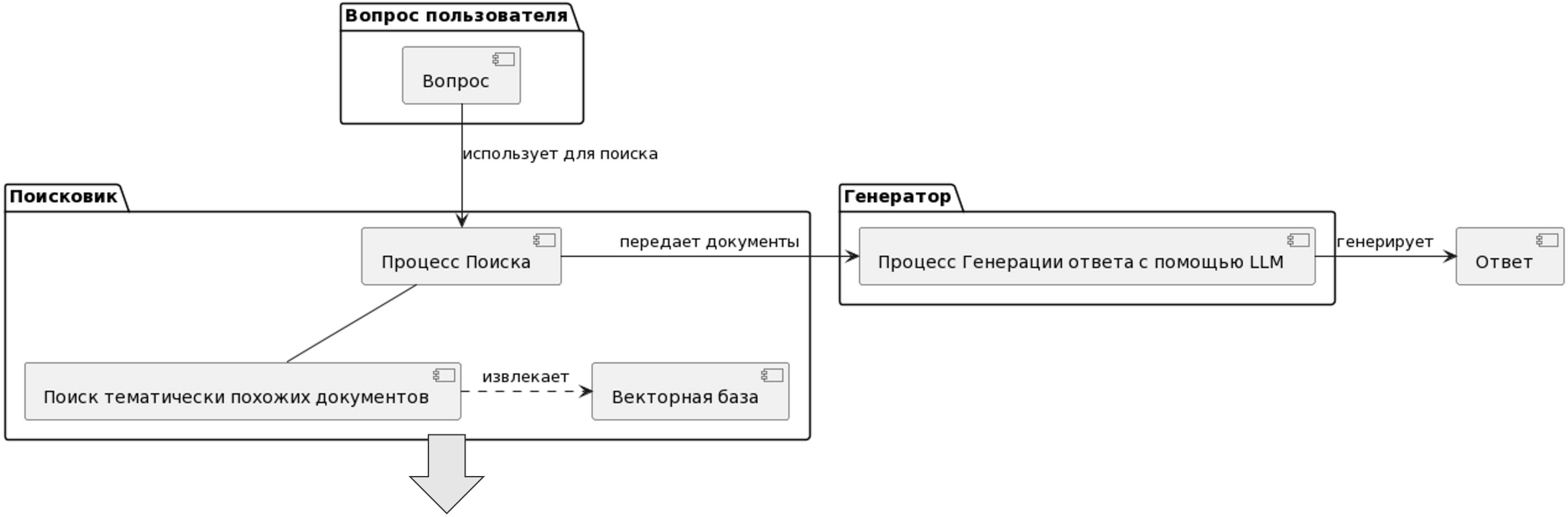
https://arxiv.org/pdf/2005.11401

RAG concepts

- **Indexing:** Splits documents into fragments and builds a vector index with an encoder.
- **Retrieval:** Finds relevant document fragments based on semantic similarity.
- **Generation:** Produces answers based on retrieved context.



RAG concepts



Possible **knowledge sources** include:

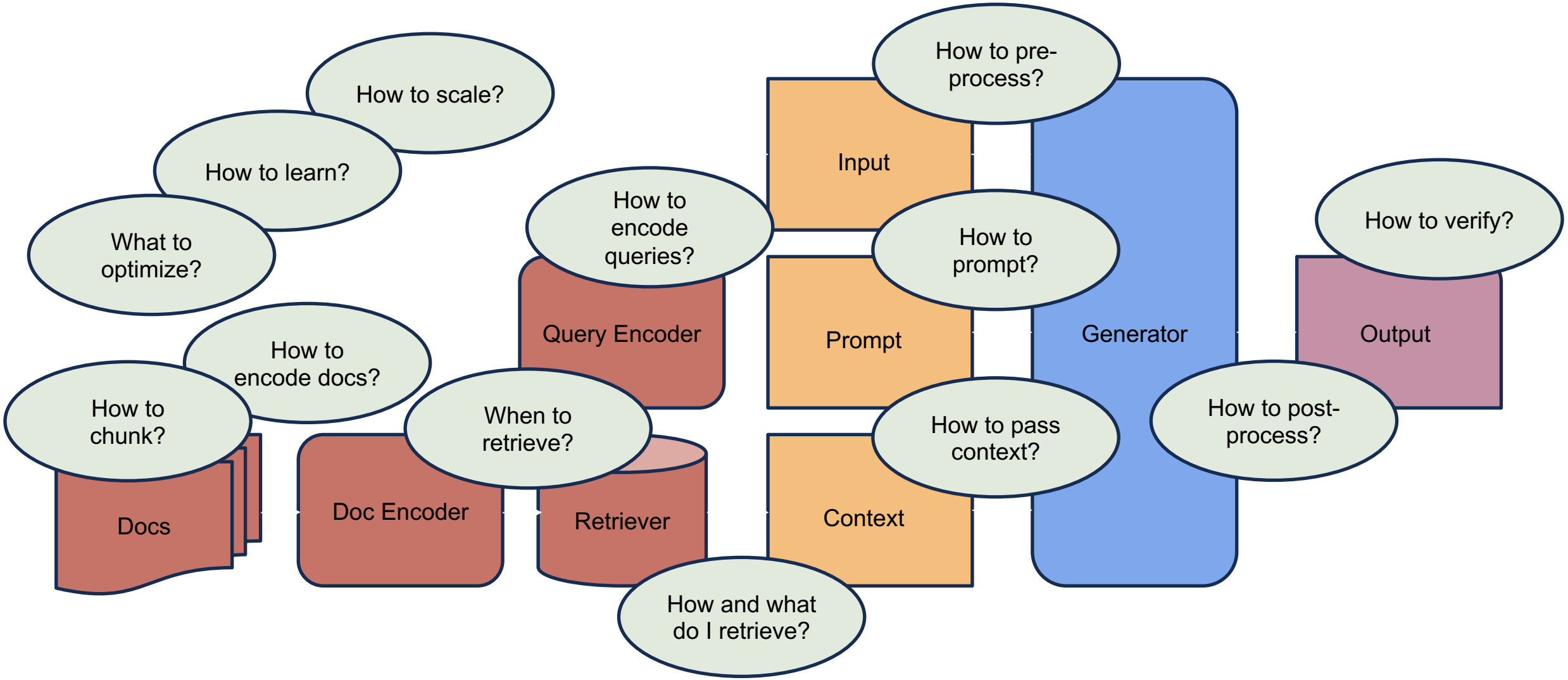
- **Structured and unstructured databases**
- **Text documents**: PDF, DOC, TXT, JSON, etc.
- **Images and video/audio files**
- **Web search**: Google, Yandex, Bing, and others…

RAG concepts

Possible model choices include:

- **OpenAI or Google** solutions **via API**
- **Open-source LLMs** such as DeepSeek, LLaMA, or Qwen
- And for the truly **ambitious** build your **own model** :)

Not as simple as it may sound…



Retrieval

Data splitting

1. Find good data sources.
2. Filter, clean, and normalize the documents.
3. Extract the text from the documents.
4. Split the text into small overlapping chunks (about 256 words) so information at boundaries isn’t lost.
5. Build a database/index from the resulting chunks.

Window size: 256 tokens
Overlap between chunks: 128 tokens (chunks overlap by ½)

The data has been chunked — how do we search now?

The data has been chunked!

How do we search now?

How IR works?

Methods:

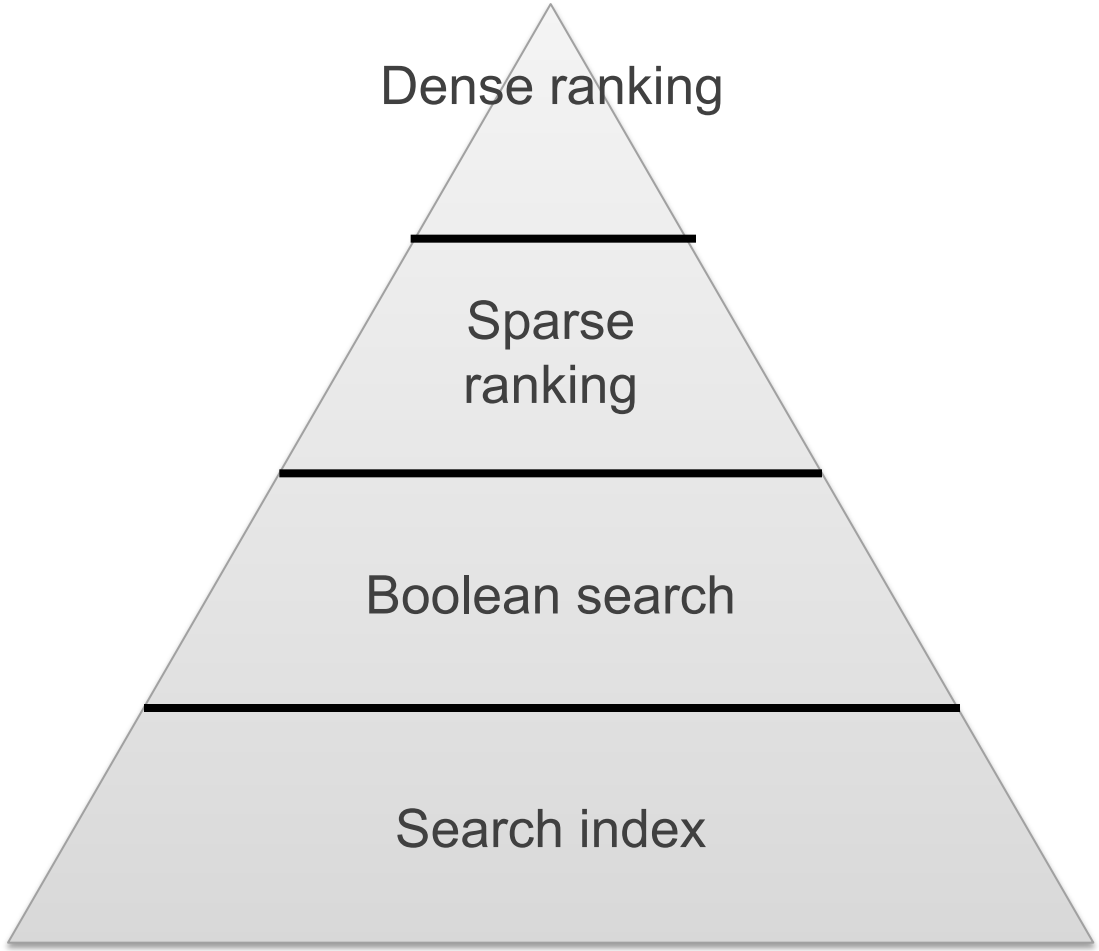
- Boolean search

Sparse vectors:

- TF-IDF
- BM25

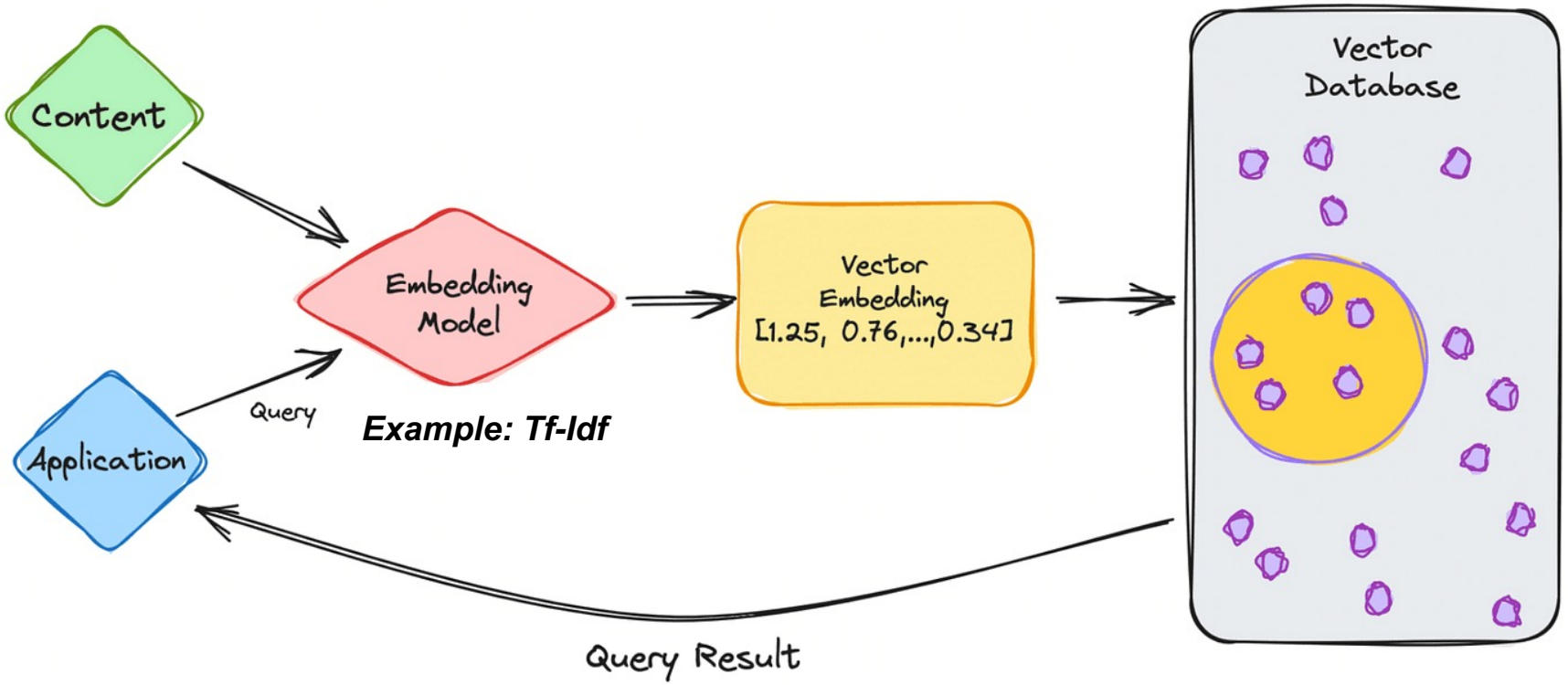
Dense vectors:

- encoders



Vector database

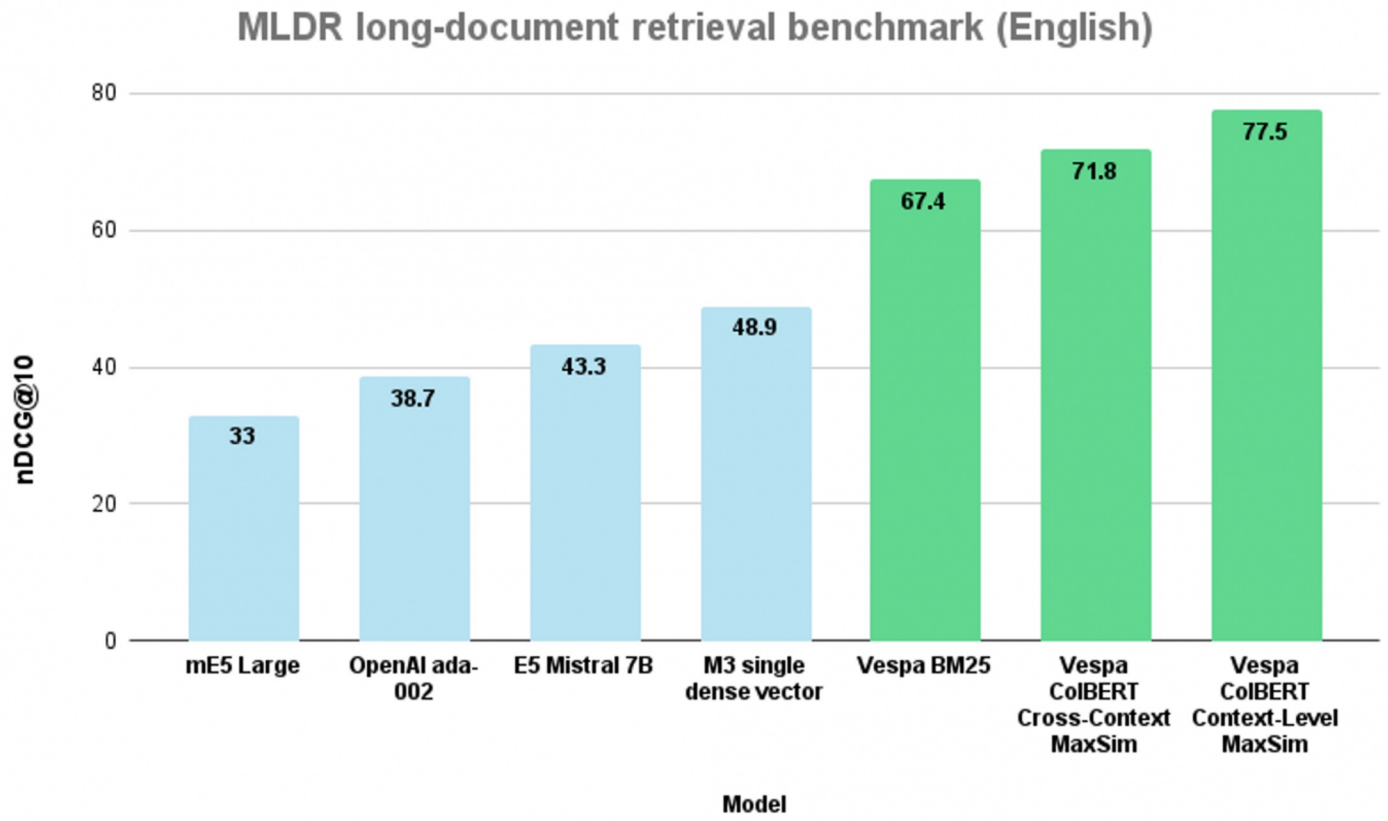
Can efficiently find approximate nearest neighbors.



Vector database

Sparse vectors:

- TF-IDF
- BM25



FAISS (Johnson et al 2017)

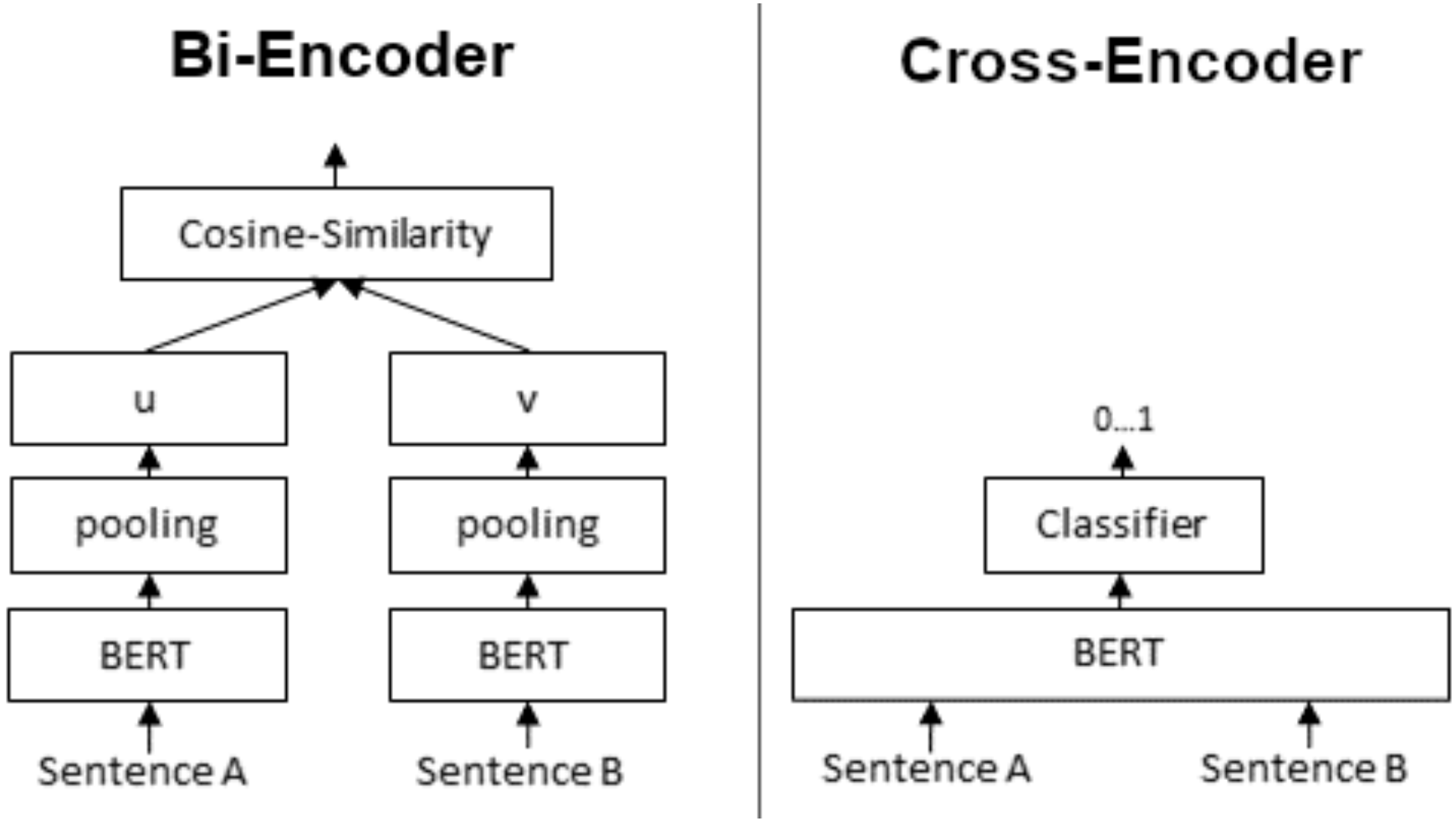
Billion-scale similarity search with GPUs

Jeff Johnson
Facebook AI Research
New York

Matthijs Douze
Facebook AI Research
Paris

Hervé Jégou
Facebook AI Research
Paris

Encoder types



Encoder types

Bi-Encoders

Pros:

- Allows for caching - count everything once, then put it in the index afterward

Cons:

- Generally inferior to cross-encoder in terms of quality

Cross-Encoders

Pros:

- Generally superior to bi-encoder in terms of quality
- Good rerankers

Cons:

- No caching

Dense Passage Retrieval (DPR)

DPR is a bi-encoder model which uses different encoders for passages $E_P(.)$ and queries $E_Q(.)$.

Both encoders are BERT-based.

Use dot-product similarity to find similar documents.

$$\text{sim}(q, p) = E_Q(q)^\intercal E_P(p)$$

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin[*], **Barlas Oğuz**[*], **Sewon Min**[†], **Patrick Lewis**,
Ledell Wu, Sergey Edunov, Danqi Chen[‡], **Wen-tau Yih**
Facebook AI [†]University of Washington [‡]Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

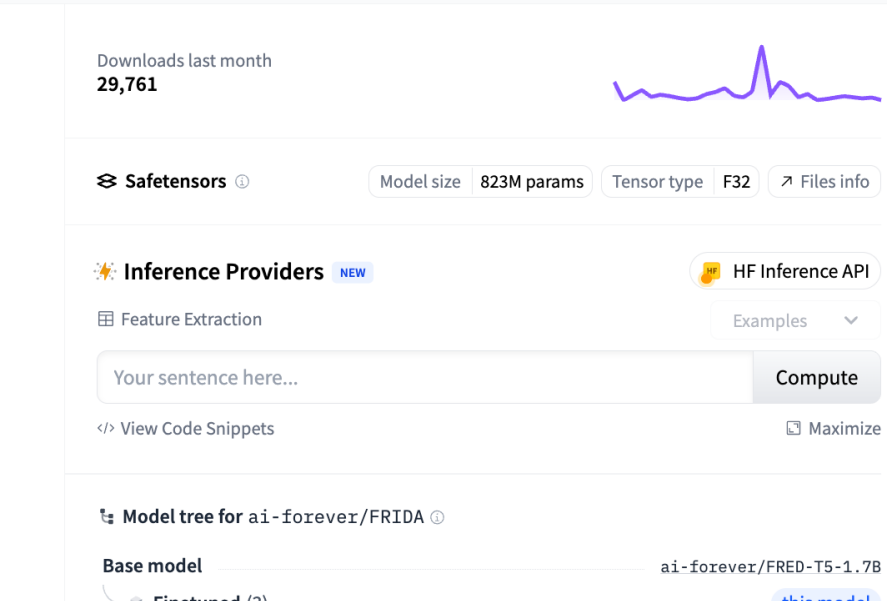
https://arxiv.org/abs/2004.04906

Abstract

Retrieval in open-domain QA is usually imple-

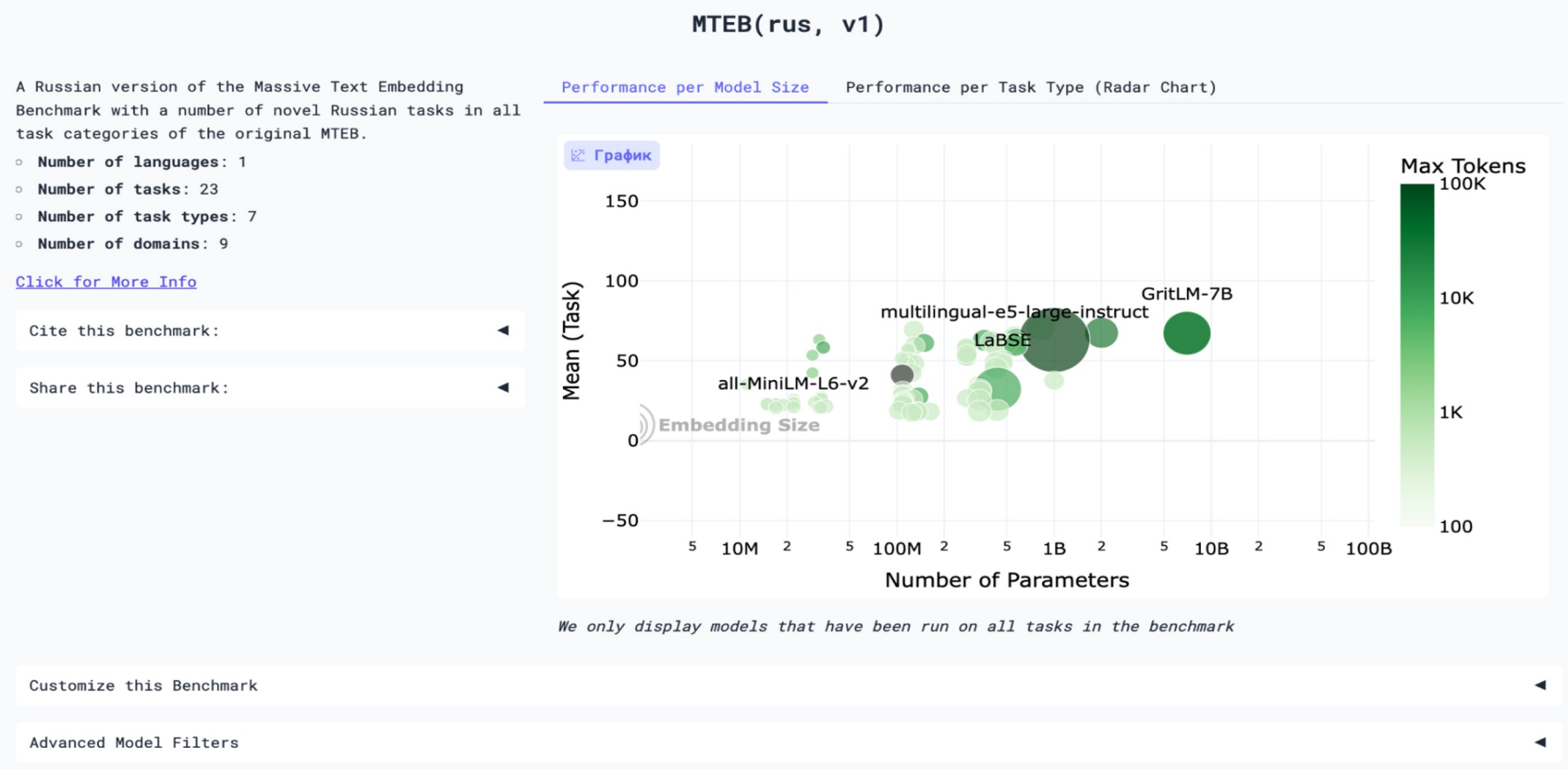
Transformer-based embedders

Encoder models are widely used for text embeddings.



https://huggingface.co/ai-forever/FRIDA

How to find a good Russian embedder? Use the ruMTEB benchmark!



How to find a good Russian embedder? Use the ruMTEB benchmark!

Summary | Performance per task | Task information

Filter...

Rank (Bo…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Classification	Clust
1	FRIDA	30%	3141	823M	1536	512	**71.06**	**68.98**	73.71	**67.49**
2	BERTA	30%	489	128M	768	512	69.34	67.86	71.22	64.78
3	Linq-Embed-Mistral	91%	13563	7B	4096	32768	67.46	64.62	71.16	62.92
4	GritLM-7B	91%	13813	7B	4096	4096	67.67	65.08	69.92	64.30
5	e5-mistral-7b-instruct	91%	13563	7B	4096	32768	67.16	64.99	69.07	64.24
6	Giga-Embeddings-instruct	⚠ NA	9649	2B	2048	32768	67.49	64.02	**73.75**	61.79
7	multilingual-e5-large-instruct	91%	1068	560M	1024	514	65.00	63.36	66.28	63.13
8	SFR-Embedding-Mistral	86%	13563	7B	4096	32768			69.81	64.92
9	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768			72.97	66.37
10	jina-embeddings-v3	100%	1092	572M	1024	8194	63.45	60.93	65.24	60.90
11	SFR-Embedding-2_R	86%	13563	7B	4096	32768			70.20	64.87

Download Table

Frequently Asked Questions ◀

Retrieval-Augmented Architectures

Training and inference

Training:

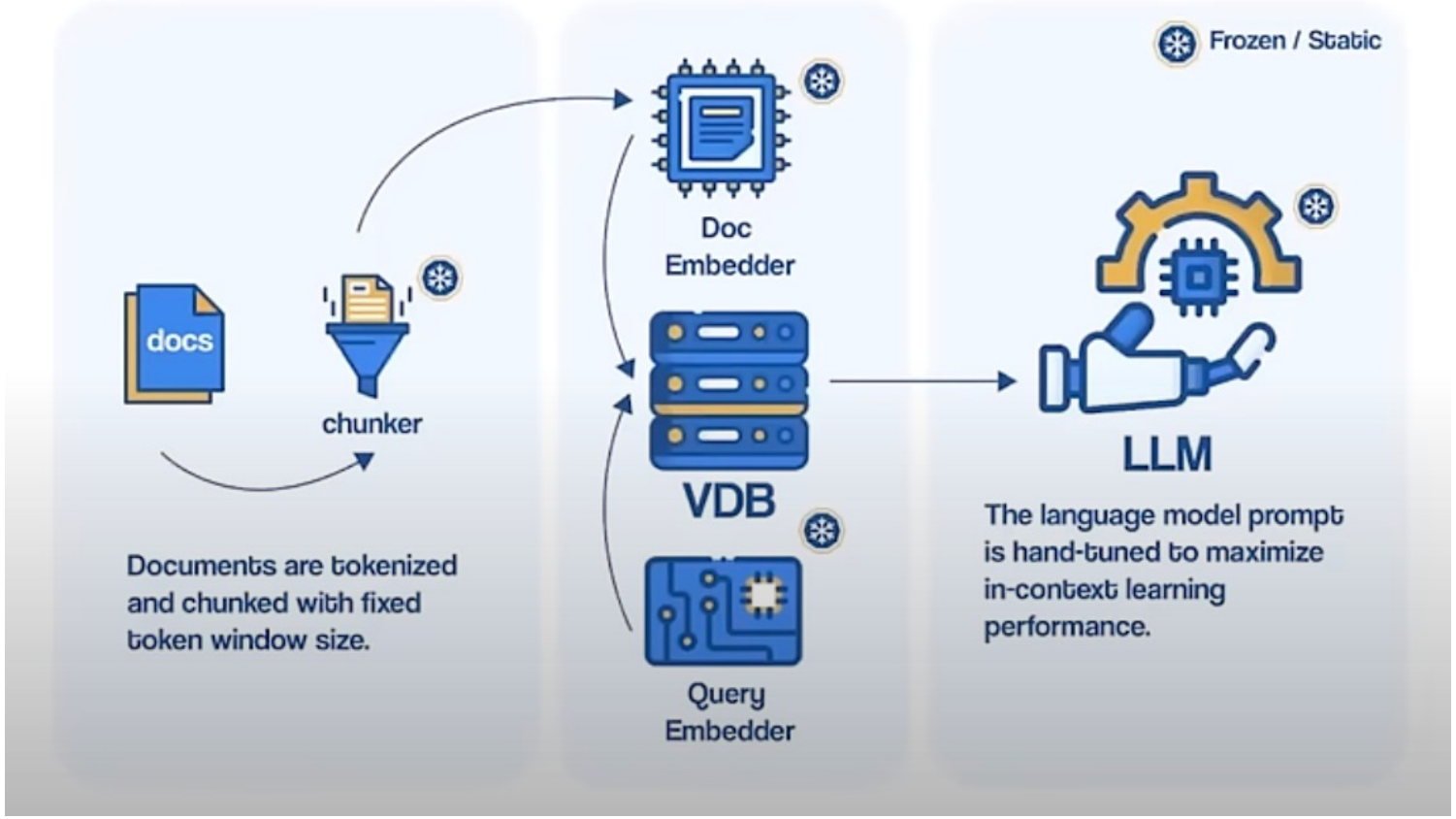
- generator?
- query encoder?
- document encoder?
- from scratch or fine-tune?

Inference:

- same or another index?

Frozen RAG

No training, only «In-Context Learning»



Frozen RAG

No training,

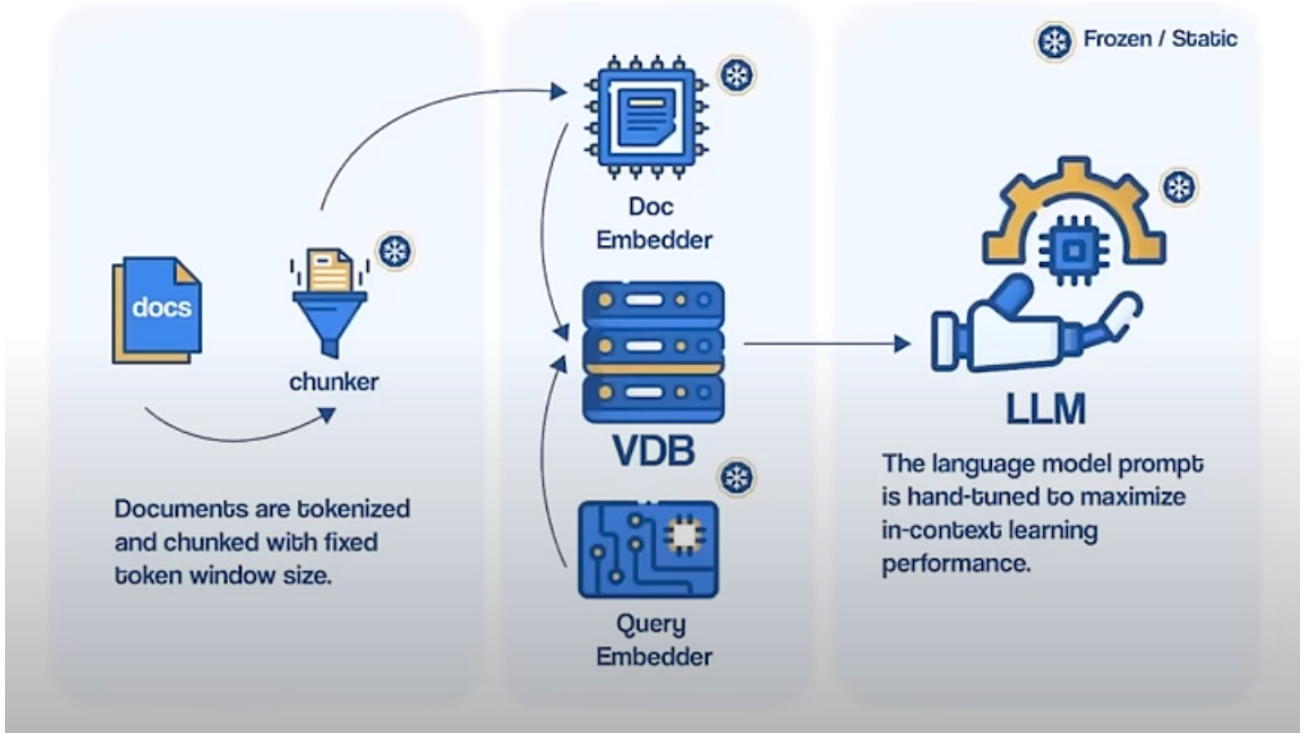
only «In-Context Learning»

Pros:

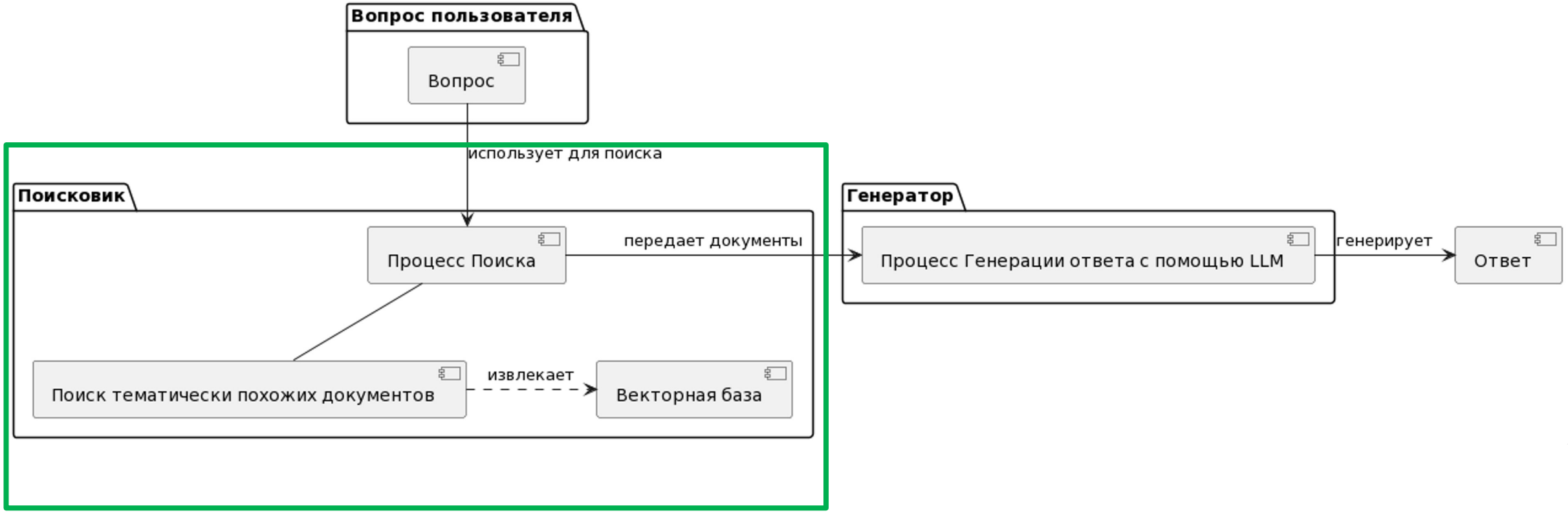
- Reasonable baseline.
- Can use black-box models for the retriever and the generator.

Cons:

- Not flexible enough.
- The resulting may not be sufficient.



Train retriever for generator



Train retriever for generator

RePlug (Shi et al 2023): train retriever, not the generator.

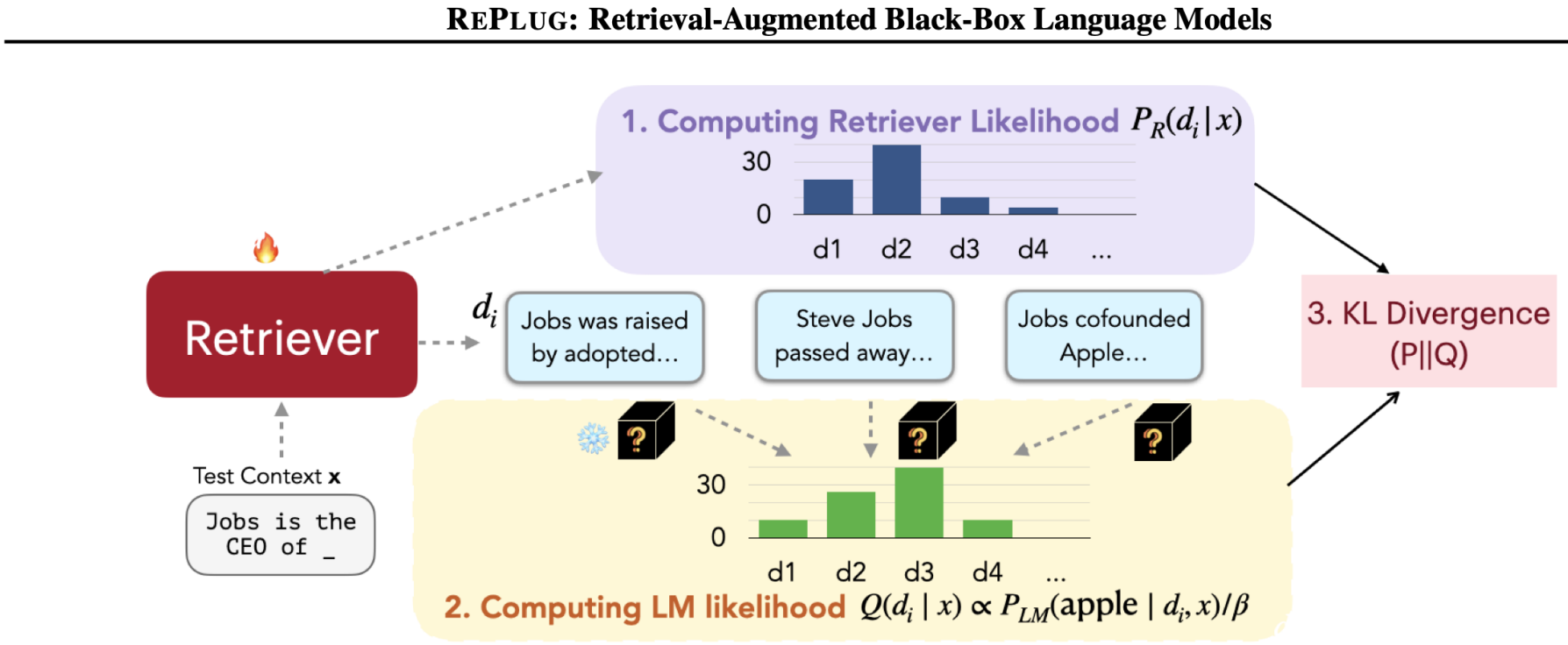


Figure 3. **REPLUG LSR training process (§4).** The retriever is trained using the output of a frozen language model as supervision signals.

Train retriever for generator

Pros:

- Retriever is fine-tune for the particular generator.
- May use black-box LLMs, such as ChatGPT, as a generator.

Cons:

- Good, but we want better!

Train retriever, not the generator.

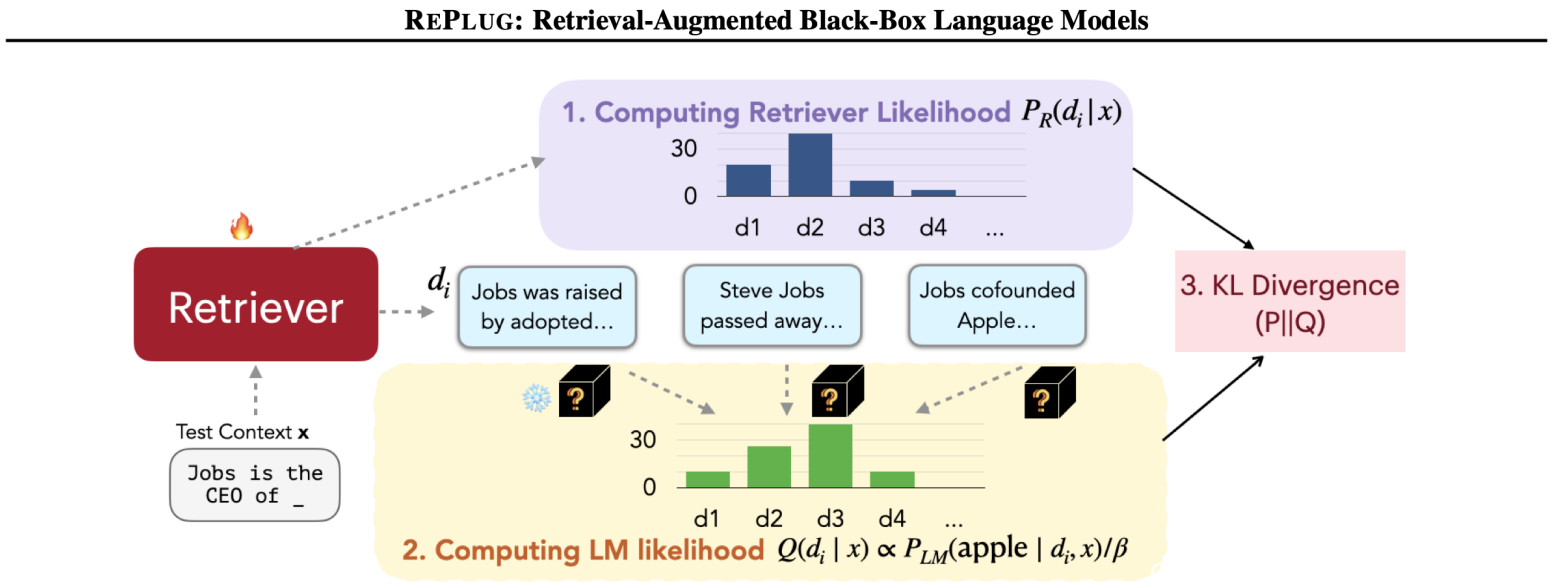
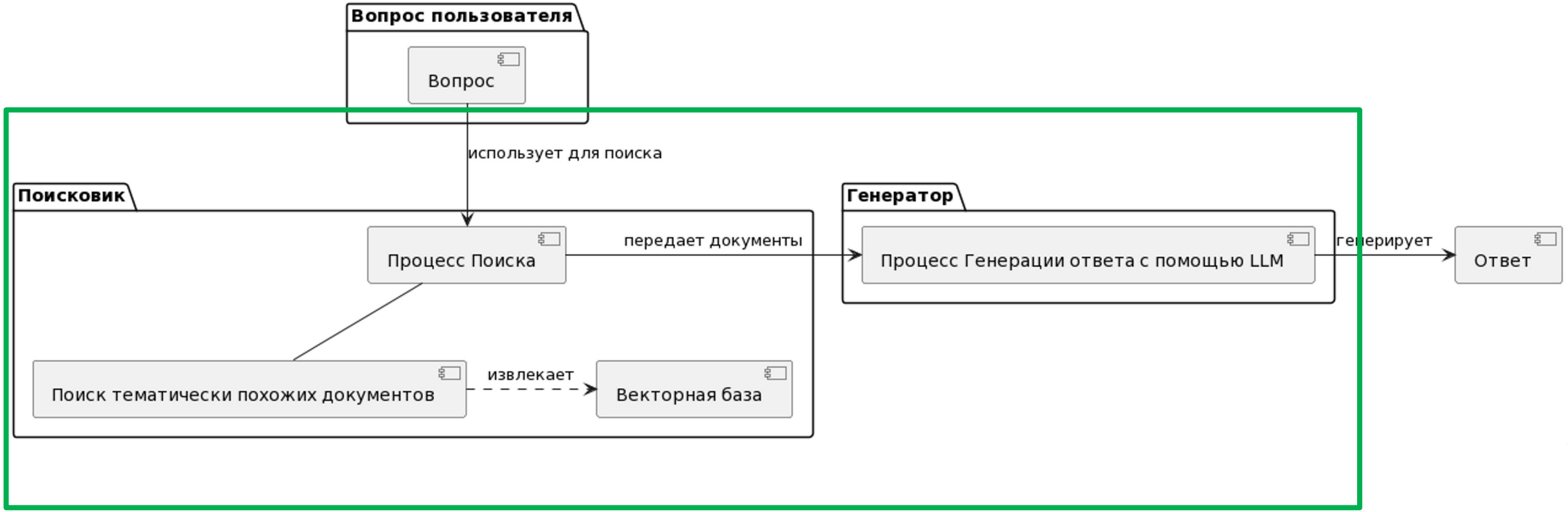


Figure 3. **REPLUG LSR training process (§4).** The retriever is trained using the output of a frozen language model as supervision signals.

Train both


Вопрос пользователя
Вопрос
использует для поиска
Поисковик
Процесс Поиска
передает документы
Генератор
Процесс Генерации ответа с помощью LLM
генерирует
Ответ
Поиск тематически похожих документов
извлекает
Векторная база

Original RAG approach

Combines a pre-trained retriever (Query Encoder + Document Index) with a pre-trained seq2seq model (Generator) fine-tuned end-to-end.

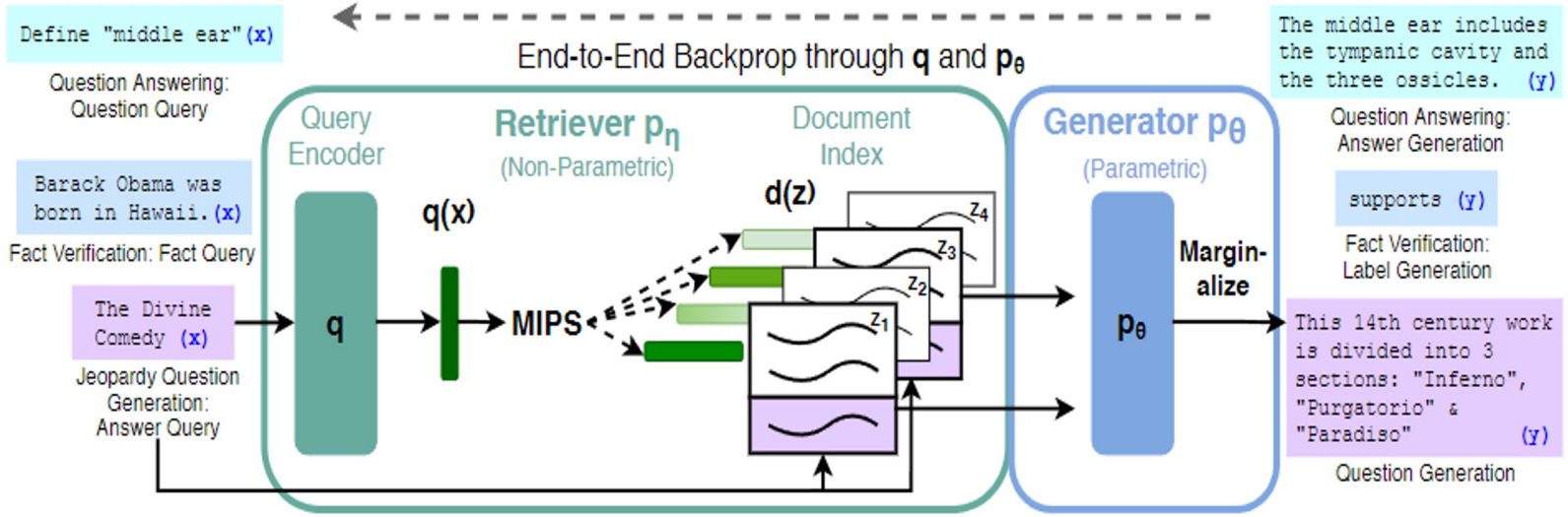


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder + Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat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

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks - NeurIPS **2021**

https://arxiv.org/pdf/2005.11401

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], **Ethan Perez**[⋆],

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

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.

1 Introduction

RAG details

a retriever $p_\eta(z|x)$

a generator $p_\theta(y_i|x, z, y_{1:i-1})$

Two models:

- **RAG-Sequence** uses the same retrieved document to generate the complete sequence.

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x, z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x, z, y_{1:i-1})$$

- **RAG-Token** can draw a different latent document for each target token and marginalize accordingly.

$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x, z, y_{1:i-1})$$

https://arxiv.org/pdf/2005.11401

RAG details

a retriever $p_\eta(z|x)$

a generator $p_\theta(y_i|x, z, y_{1:i-1})$

Two models:

- **RAG-Sequence** uses the same retrieved document to generate the complete sequence.

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x, z, y_{1:i-1})$$

- **RAG-Token** can draw a different latent document for each target token and marginalize accordingly.

$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x, z, y_{1:i-1})$$

- Retriever is based on **DPR**.

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(z) = \text{BERT}_d(z), \;\; \mathbf{q}(x) = \text{BERT}_q(x)$$

- Generator is **a BART- large** model.

- **Training**: the retriever and generator components are **jointly trained** without any direct supervision on what document should be retrieved.

https://arxiv.org/pdf/2005.11401

Experiments & results

Takeaways:

- On all four open-domain QA tasks, RAG sets a new state of the art.
- RAG combines the generation flexibility of the “closed-book” (parametric only) approaches and the performance of "open-book" retrieval-based approaches.
-
- RAG can generate correct answers even when the correct answer is not in any retrieved document.

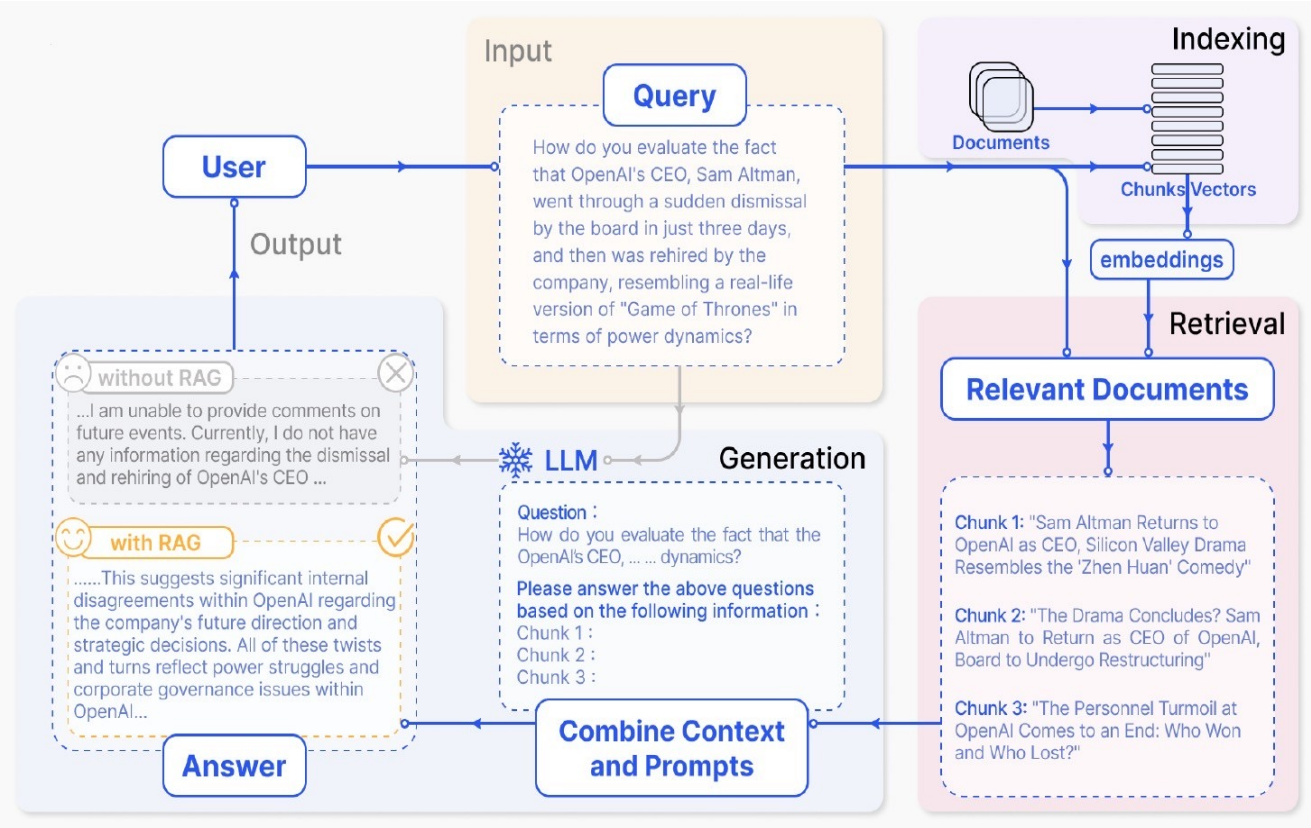
Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- /50.1	37.4	-
	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

https://arxiv.org/pdf/2005.11401

Advanced Frozen RAG

- Active community
- Frameworks: **LangChain**, LlamaIndex, GigaChain
- Vector Databases: Chroma,
- Weaviate, etc
- Cool examples:
  - *Child-Parent Retriever*
  - *Hybrid Search*
  - *LLM reranker*
  - …



Advanced RAG: extra

Advanced RAG includes additional **pre-** and **post-processing steps.**

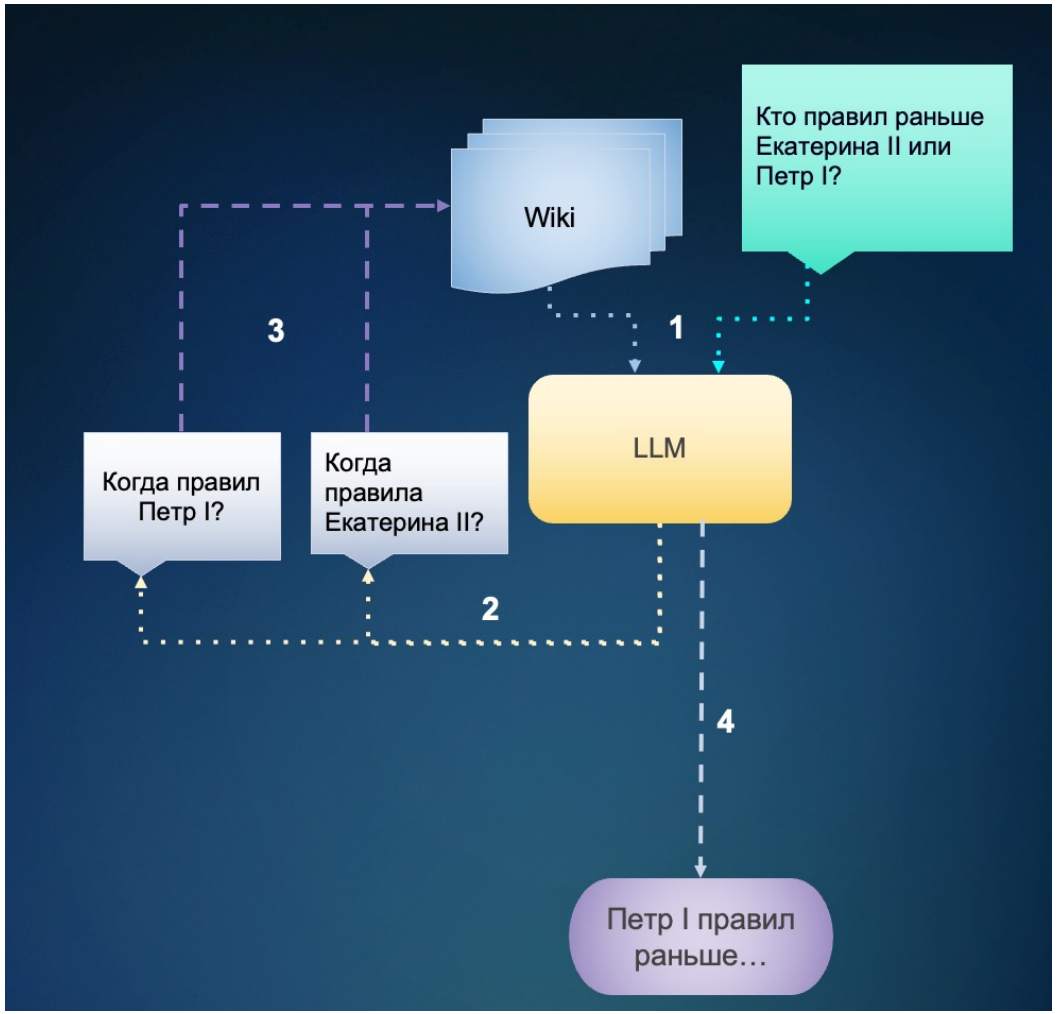
Pre-processing:

- Applying a classifier to determine whether factual information is needed to answer the question.

- Rewriting the query to add context and eliminate semantic mismatches between the question and document fragments.

Multi-hop

We can initiate decomposition of a complex query into several clarifying sub-questions.



Context

Reformulate the question to include context information and equalize semantic differences between questions and document fragments.

Какая самая длинная река в мире?

Какая самая длинная река в мире?

Самая длинная река в мире — это река **Нил** или **Амазонка**, в зависимости от метода измерения.

- **Нил** традиционно считается самой длинной рекой, с длиной около **6 650 километров**. Он протекает через 11 стран Африки, включая Египет, Судан и Эфиопию.
- **Амазонка**, которая находится в Южной Америке, имеет длину около **6 992 километров** по некоторым современным исследованиям. Однако это значение иногда оспаривается из-за сложности точного определения истока реки.

Таким образом, первое место зависит от подхода и методики измерения.

*Сколько у **них** притоков?*

Сколько притоков у реки Амазонка?

Река Амазонка

- У Амазонки **более 1 100 притоков**, из которых около **17 считаются крупными** (длина превышает 1 500 км). Среди крупнейших притоков:
  - **Мадейра**,
  - **Риу-Негру**,
  - **Журуа**,
  - **Пурус**,
  - **Тапажос**.

Бассейн Амазонки — крупнейший в мире, и её притоки образуют сложную и густую сеть благодаря влажному климату тропиков.

Таким образом, река Амазонка имеет значительно больше притоков, чем Нил, благодаря особенностям тропического климата и обширной площади бассейна.

Сколько притоков у реки Нил?

Количество притоков у рек Нила и Амазонки значительно отличается из-за географических особенностей и размеров их бассейнов.

Река Нил

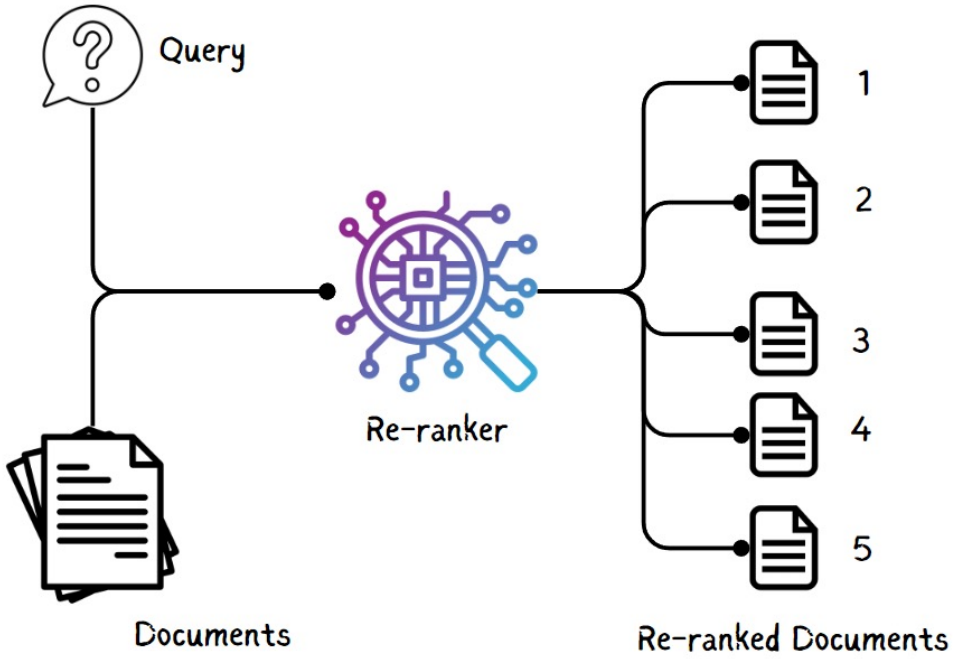
- У Нила **около 140 притоков**, но основные притоки — это:
  - **Белый Нил** (исток в Восточной Африке, озеро Виктория),
  - **Голубой Нил** (исток в Эфиопии, озеро Тана),
  - **Атбара** (меньший приток из Эфиопии).

Advanced RAG: extra

The advanced RAG paradigm includes additional processing before retrieval and after retrieval.

After retrieval:

- Rank the corpus of the retrieved documents.
- Filter the retrieved context and compress it to reduce the length of the context window.



What else? Prompting

A good System Prompt can boost quality by at least +2%.

Prompt

Ты - система генерации ответа на основе поисковой выдачи. Тебе дан вопрос и поисковая выдача.
Создай краткий и информативный ответ на заданный вопрос.
{LAST_MESSAGES}
QUESTION: {QUESTION}
========
{{SEARCH_RESULTS}}
=========
FINAL ANSWER:

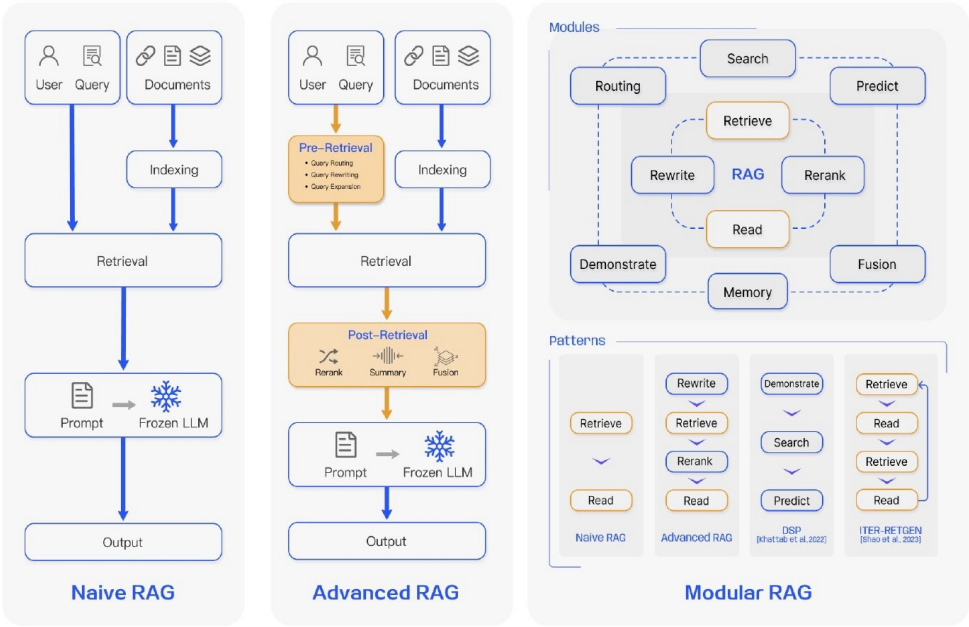
SEARCH_RESULTS

{{SOURCE_TEXT}}
{{SOURCE_INDEX}}

Где {{SOURCE_TEXT}} – текст документа из поисковой выдачи
А {{SOURCE_INDEX}} – индекс документа из поисковой выдачи

Modular RAG

- Structurally, it is more flexible, with more specific functional modules such as **query search engines** and **multiple response pooling.**
- Technologically, it combines **search with fine-tuning**, r**einforcement learning** and other methods.



RAG Evaluation

Evaluation of RAG systems

Critical features of good RAG evaluation:

- **Realism:** quality correlation with real scenarios.
- **Richness:** diverse set of instances. (common and complex, use cases...)
- **Insightfulness:** (interpretability) understanding of performance on different slices of the data.
- **Reliability:** accurate GT, the metrics capture the model performance well, statistical significance.
- **Longevity:** long term data, refreshed data.

Table 3: Summary of evaluation frameworks

Evaluation Framework	Evaluation Targets	Evaluation Aspects	Quantitative Metrics
RGB†	Retrieval Quality Generation Quality	Noise Robustness Negative Rejection Information Integration Counterfactual Robustness	Accuracy EM Accuracy Accuracy
RECALL†	Generation Quality	Counterfactual Robustness	R-Rate (Reappearance Rate)
RAGAS‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * Cosine Similarity
ARES‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	Accuracy Accuracy Accuracy
TruLens‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * *

Retrieval metrics

Reranking metrics: retriever can be evaluated with common ranking metrics (*NDCG, MRR, MAP, etc.*).

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k} + \text{false positives@k})}$$

$$\text{Context Recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of sentences in GT}|}$$

Entity-based: based on fine-grained context analysis. These metrics evaluate overlapping between entities in context (named entities or sentences).

$$\text{context relevancy} = \frac{|\text{Relevant sentences in retrieved context}|}{|\text{Total number of sentences in retrieved context}|}$$

$$\text{context entity recall} = \frac{|CE \cap GE|}{|GE|}$$

Generation metrics

Direct comparison with Ground Truth answers by EM or similarity.

Any distance in embedding space or string editing distances could be used as similarity.

Models	Accuracy	
	Retrieved Chunk	Ground-truth Chunk
GPT-4	**0.56**	**0.89**
ChatGPT	0.44	0.57
Llama-2-70b-chat-hf	0.28	0.32
Mixtral-8x7B-Instruct	0.32	0.36
Claude-2.1	0.52	0.56
Google-PaLM	0.47	0.74

Table 6: Generation accuracy of LLMs.

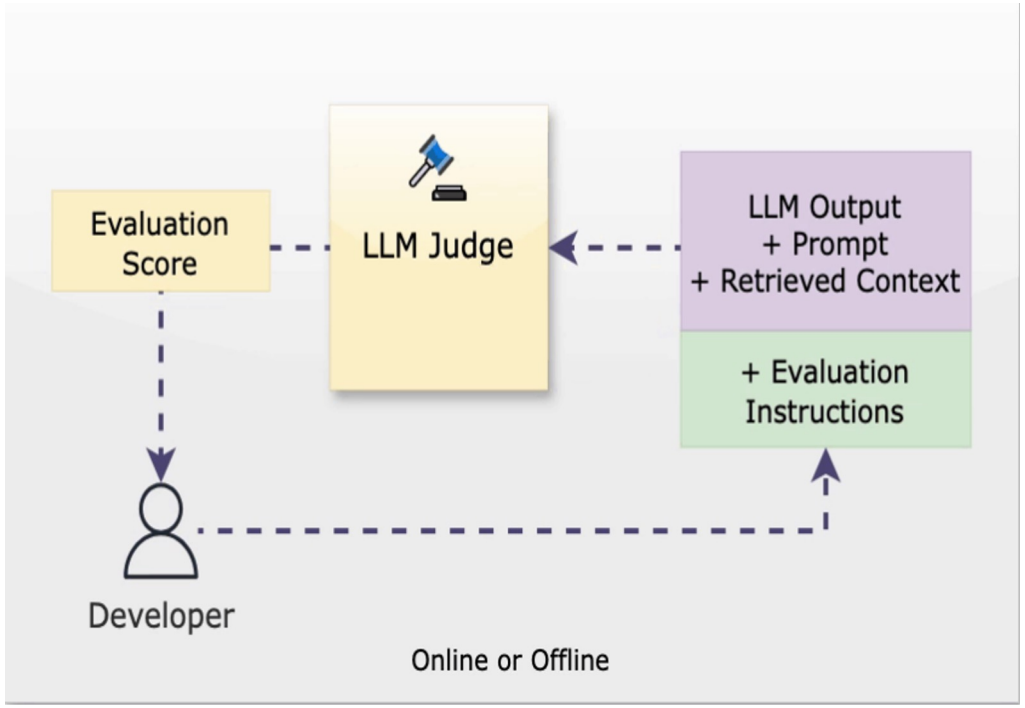
Generation metrics

LLM as a Judge

Categories: *Ideal, Acceptable, Missing, Incorrect answer.*
Each response is compared with the ground truth.

Quality is evaluated across specific aspects:

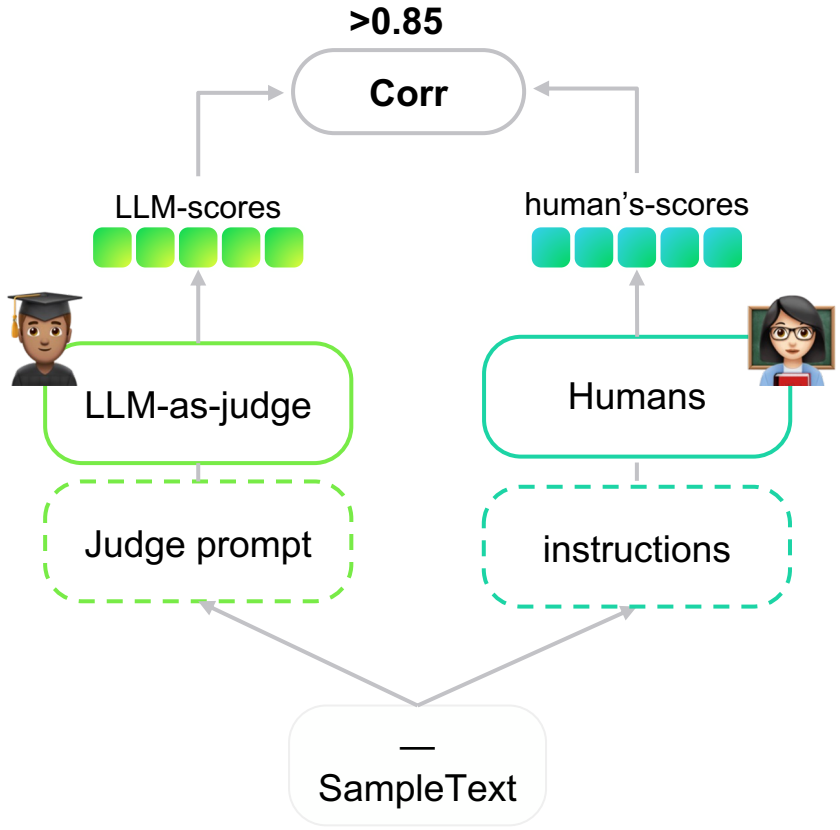
- Harmlessness
- Correctness
- Maliciousness
- Coherence
- Conciseness



Generation metrics

Limitations:

- LLM judges tend to favor (i.e., overrate) answers from similar models with similar behavior, specific stylistic patterns, or certain lengths (too long or too short).
- You cannot use the same model as both the RAG reasoner and the judge — not even from the same model family, and not even after SFT.
- The judge's instruction must demonstrate **a strong correlation with human annotations.**



Domain-specific metrics

Some domains allows to build specific metrics for answer evaluation.

For example:

- *In code domain we can use not only EM on the whole prediction but also the Identity Matching.*
- *Also we can run Unit Tests on the predicted snippet and obtain the real quality of the answer.*

ID	N.	Oracle	In-File	RepoCoder Iterations			
				1	2	3	4
1.	67	56.72	29.85	53.73	**55.22**	55.22	55.22
2.	146	43.84	27.40	41.78	43.84	**44.52**	44.52
3.	64	32.81	10.94	25.00	**34.38**	31.25	32.81
4.	32	34.38	28.13	34.38	**37.50**	34.38	34.38
5.	22	40.91	31.82	31.82	**36.36**	31.82	36.36
6.	42	38.10	9.52	28.57	**38.10**	38.10	38.10
All	373	42.63	23.32	38.34	**42.63**	41.82	42.36

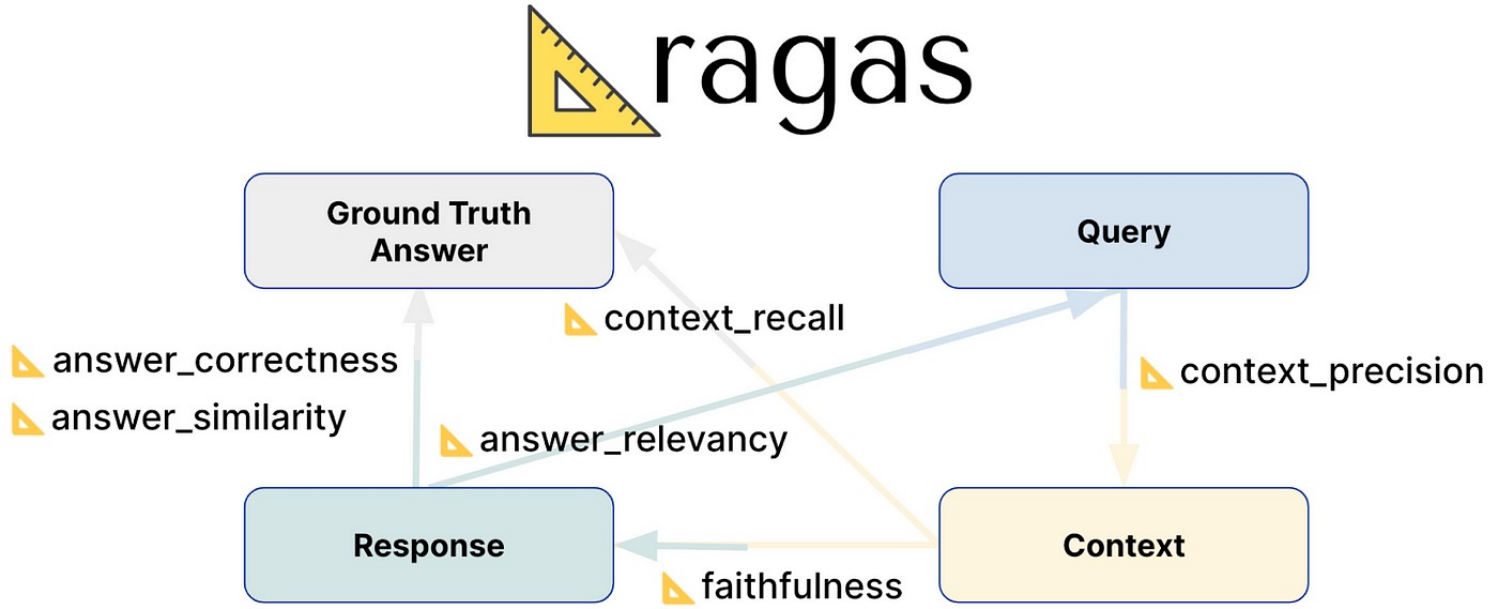
Table 3: Performance comparison on the function body completion dataset using GPT-3.5-Turbo. Results display the Pass Rate (PR) of each method as evaluated using test cases. Numbers are presented in percentage (%), with the best performance highlighted in bold. *ID* represents the repository IDs, and *N.* indicates the number of test samples in each repository.

https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

Ragas

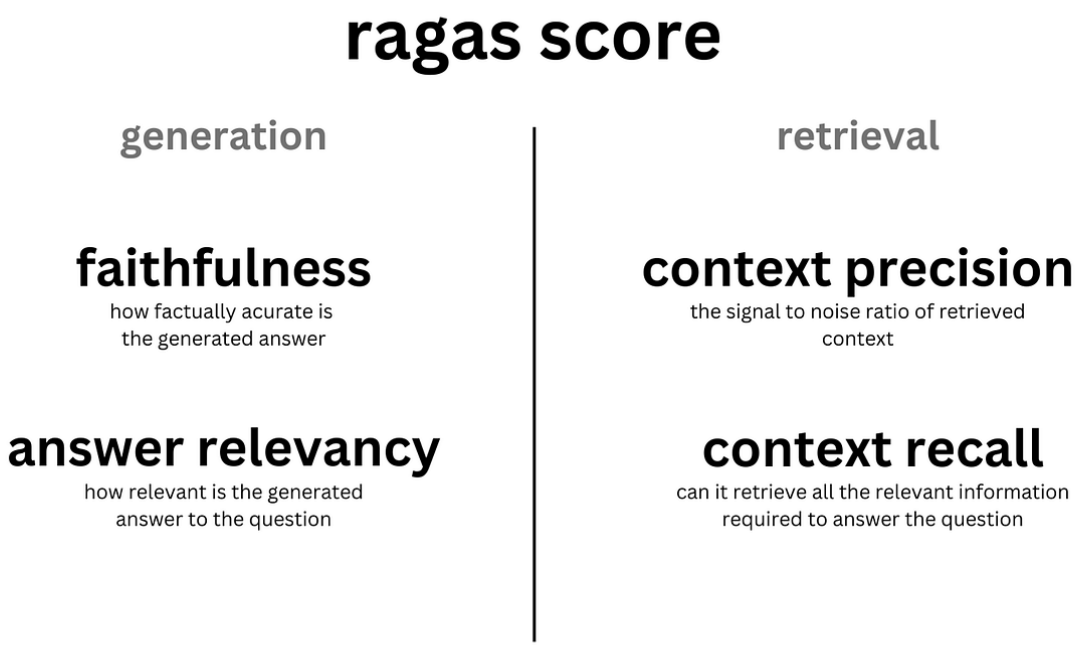
Ragas is a **framework designed to evaluate RAG systems**, which combine language models LLMs with external knowledge retrieval mechanisms.

Provides a structured approach to evaluate **both the generator and retriever components** of a RAG pipeline through a set of metrics that can be applied without requiring extensive labeled datasets.



https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

Key metrics of Ragas



https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

Key metrics of Ragas

Faithfulness: *the factual accuracy of the generated answer*s by checking if the statements made in the answers are supported by the provided context. It is computed by analyzing the validity of each statement derived from the generated answer against the context.

Answer Relevancy: Measures *how relevant* and directly related the generated answer is to the posed question. It evaluates the similarity between probable questions that could elicit the same answer and the actual question asked.

The Answer Relevancy is defined as the mean cosine similarity of the original `user_input` to a number of artifical questions, which where generated (reverse engineered) based on the `response`:

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N} cos(E_{g_i}, E_o)$$

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N} \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\|\|E_o\|}$$

https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

Key metrics of Ragas

Context Precision: *Evaluates the signal-to-noise ratio within retrieved contexts*, determining how many sentences from the context are necessary to answer the question compared to the total number of sentences retrieved.

Context Recall: Checks whether all necessary information needed to support the generated answer can be found in the retrieved context. It compares statements from a ground truth answer against what has been retrieved.

Context Precision is a metric that measures the proportion of relevant chunks in the `retrieved_contexts`. It is calculated as the mean of the precision@k for each chunk in the context. Precision@k is the ratio of the number of relevant chunks at rank k to the total number of chunks at rank k.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k} + \text{false positives@k})}$$

Where K is the total number of chunks in `retrieved_contexts` and vk∈{0,1} is the relevance indicator at rank k.

Computed using `user_input`, `reference` and the `retrieved_contexts`, and the values range between 0 and 1, with higher values indicating better performance. This metric uses `reference` as a proxy to `reference_contexts` which also makes it easier to use as annotating reference contexts can be very time consuming. To estimate context recall from the `reference`, the reference is broken down into claims each claim in the `reference` answer is analyzed to determine whether it can be attributed to the retrieved context or not. In an ideal scenario, all claims in the reference answer should be attributable to the retrieved context.

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

https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

DRAGON. Idea

The first dynamic benchmark for evaluating RAG systems in the news domain in Russian.

Focuses on working with **external knowledge sources** and retrieving **up-to-date information.**

*Developed in collaboration with ITMO, MISIS, HSE University, and MWS AI.

DRAGON. Motivation

What problem does it solve?

It **addresses the issue where RAG test data leaks into model pretraining:**

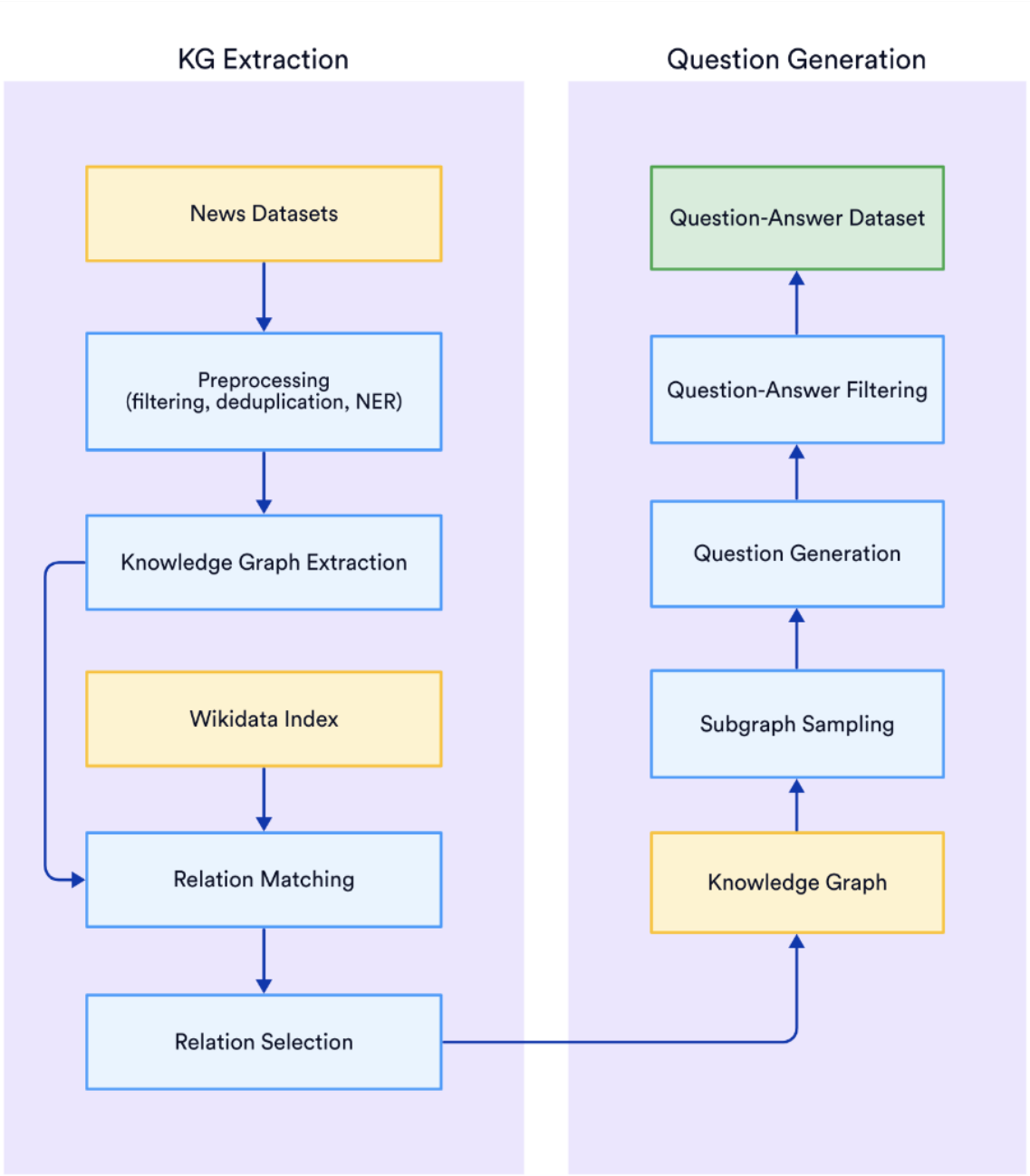
- The model is trained on data up to a certain date.
- But the benchmark uses always up-to-date news.
- Regular benchmark updates ensures data freshness and fairness.

Question generation

Automatic question generation in three stages:

- **Knowledge extraction.** A knowledge graph in the Wikidata style is built from a news corpus.
- **Question generation.** Parts of the graph are passed to a language model to create QA pairs.
- **Filtering.** Basic filters, NER, POLLUX.

Result: Structure + flexibility = high-quality and diverse questions from real-world data.



Leaderboard

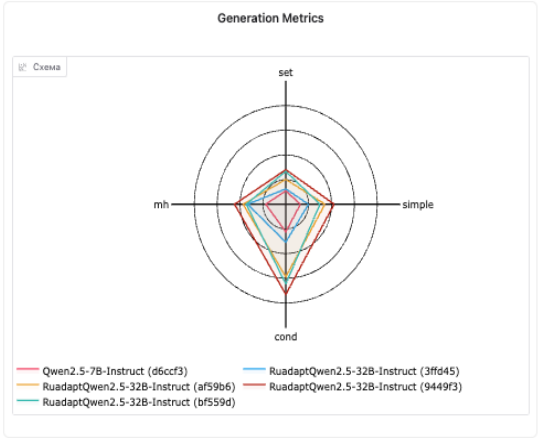
Baselines: a range of popular RAG systems.

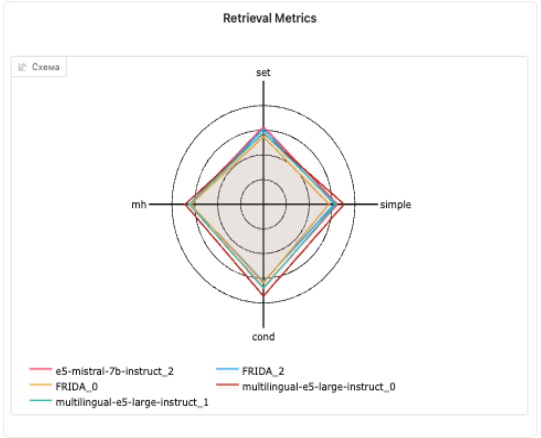
Always up to date: scores are recalculated with every dataset release.

Simple evaluation: easy to compare your system against standard baselines.

Result: more transparency, engagement, and healthy competition.

Faster progress in RAG!





Clear Charts

Overall Table | By Question Type

Model	Embeddings	Top k	Retrieval (avg)	Generation (avg)	Total Score	Version	Last Updated
RuadaptQwen2.5-32B-Instruct (9449f3)	multilingual-e5-large-instruct_0	2	0.6769	0.4702	0.5736	1.11.0	2025-07-20
RuadaptQwen2.5-32B-Instruct (bf559d)	multilingual-e5-large-instruct_1	5	0.6362	0.3892	0.5127	1.11.0	2025-07-20
RuadaptQwen2.5-32B-Instruct (af59b6)	FRIDA_0	5	0.5982	0.3762	0.4872	1.11.0	2025-07-20

Full Pipeline

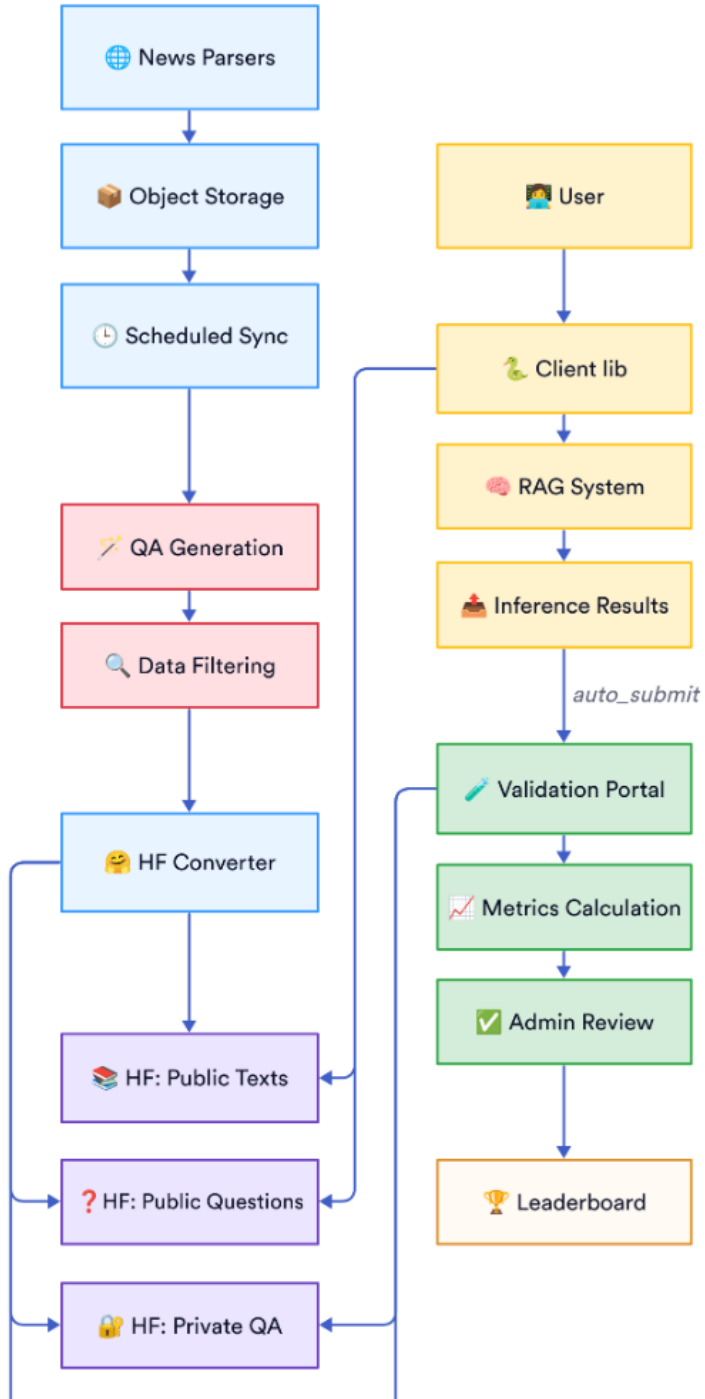
Full pipeline: Question generation + ready-to-use infrastructure for system evaluation.

Features:

- Run inference with any RAG pipeline locally
- Perform automatic answer evaluation via LLM-as-Judge
- Submit results to the leaderboard for comparison

Result:

- Simplifies evaluation: everything is reproducible, transparent, and research-friendly
- Opens up vast possibilities for evaluating RAG systems in a dynamic setting: research papers, posts, blogs, etc.



https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

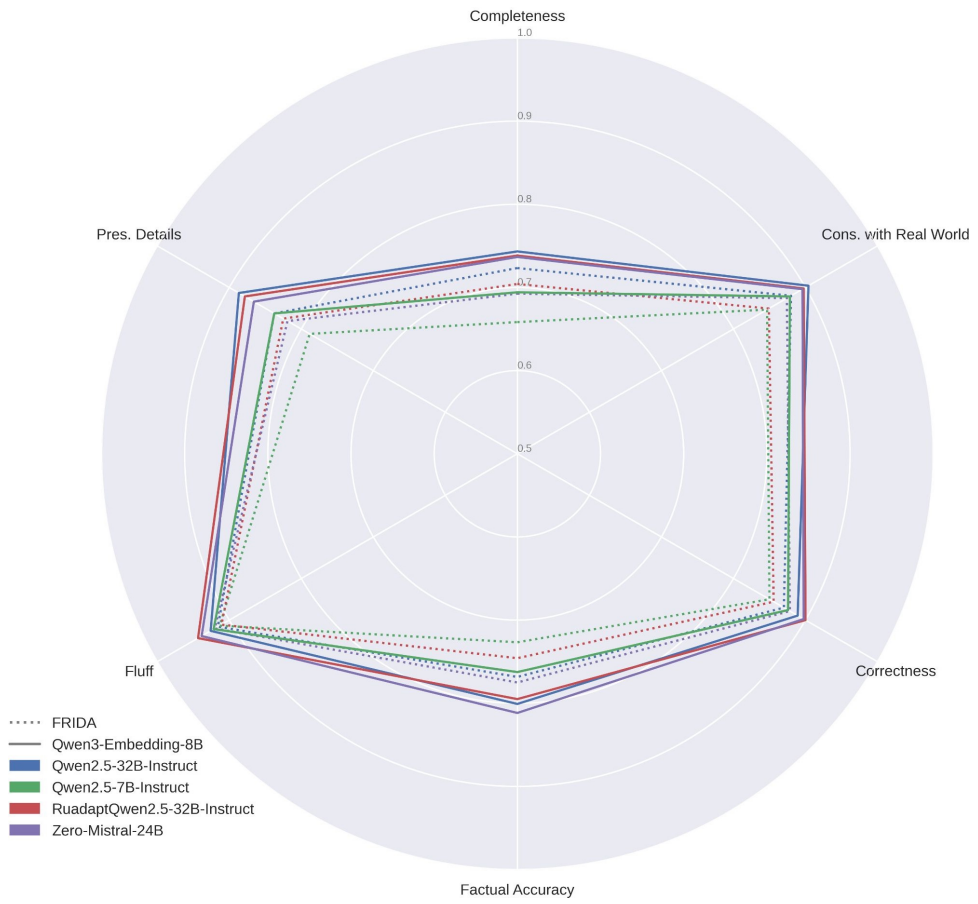
Retrieval

Retriever	Hit Rate	Recall	NDCG
FRIDA	0.88	0.84	0.85
$mE5_{\text{Large Instruct}}$	<u>0.90</u>	0.87	0.86
$\text{Qwen 3}_{\text{Embedding 8b}}$	**0.93**	**0.90**	**0.89**
$\text{E5 Mistral}_{\text{7b Instruct}}$	**0.93**	<u>0.89</u>	<u>0.87</u>

Table 2: Retrieval evaluation results. Best score is in bold, second best is underlined.

https://medium.com/@danushidk507/evaluation-with-ragas-873a574b86a9

Full evaluation



Retriever	LLM	Rouge-L	SM	Judge Score
FRIDA	Qwen $2.5_{32b\ Instruct}$	0.42	0.47	0.83
	Qwen $2.5_{7b\ Instruct}$	0.33	0.43	0.80
	Ruadapt $Qwen_{32b\ Instruct}$	0.40	0.45	0.82
	Zero $Mistral_{24B}$	0.44	0.45	0.83
	Gemma $3_{12b\ it}$	0.44	0.44	0.81
	Gemma $3_{27b\ it}$	0.40	0.47	0.81
Qwen $3_{Embedding\ 8b}$	Qwen $2.5_{32b\ Instruct}$	0.44	**0.51**	**0.86**
	Qwen $2.5_{7b\ Instruct}$	0.36	0.45	0.83
	Ruadapt $Qwen_{32b\ Instruct}$	0.42	0.48	**0.86**
	Zero $Mistral_{24B}$	0.45	0.48	**0.86**
	Gemma $3_{12b\ it}$	**0.47**	0.47	0.84
	Gemma $3_{27b\ it}$	0.43	<u>0.49</u>	<u>0.85</u>
E5 $Mistral_{7b\ Instruct}$	Qwen $2.5_{32b\ Instruct}$	0.44	**0.51**	**0.86**
	Qwen $2.5_{7b\ Instruct}$	0.36	0.46	0.82
	Ruadapt $Qwen_{32b\ Instruct}$	0.41	0.47	<u>0.85</u>
	Zero $Mistral_{24B}$	0.45	0.47	<u>0.85</u>
	Gemma $3_{12b\ it}$	<u>0.46</u>	0.47	0.83
	Gemma $3_{27b\ it}$	0.41	0.48	<u>0.85</u>
$mE5_{Large\ Instruct}$	Qwen $2.5_{32b\ Instruct}$	0.43	<u>0.49</u>	<u>0.85</u>
	Qwen $2.5_{7b\ Instruct}$	0.34	0.44	0.82
	Ruadapt $Qwen_{32b\ Instruct}$	0.39	0.46	0.83
	Zero $Mistral_{24B}$	0.44	0.46	0.84
	Gemma $3_{12b\ it}$	<u>0.46</u>	0.46	0.83
	Gemma $3_{27b\ it}$	0.41	0.48	0.83

Table 3: End-to-end RAG-system evaluation results. Retrieval evaluation results. SM stands for Substring Matching. The judge's score is computed by averaging the results among the criteria. The best score is in bold, and the second-best score is underlined.

RAG conclusion

- There is no one “silver bullet”: **there is a wide variety of approaches and ideas**.
- For each use case, **you need a customized solution** based on cost/benefit considerations.
- Not only are system architectures important, but so **is the quality of the data** in the index.
- Special attention to **the evaluation of RAG systems**.