

RAG

генерация на основе базы знаний

• REC

Проверить, идет ли запись

Меня хорошо видно
&& слышно?



Ставим "+", если все хорошо
"-", если есть проблемы

Тема вебинара

RAG - генерация на основе базы знаний



Андрей Коняев

**Consultant GenAI Machine Learning Engineering
(Professional Research & Development Engineer I)**
@ T-Systems Internatinal GmbH (Deutsche Telekom)

(Ex) Data Scientist / NLP Engineer
@ text2knowledge GmbH

Телеграм:
@KONIANKO



Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в телеграмм чате группы



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара

0. Знакомство

1. Что такое RAG?

2. Зачем нужен RAG в NLP сервисах?

3. Области применения RAG

4. Типы RAG и пример архитектуры сервиса с RAG

5. RAG vs Fine-Tuning

6. Оценка работы сервиса с RAG

7. Демонстрация QA системы с RAG

Цели вебинара

К концу занятия вы сможете

1. Объяснить подход RAG
 2. Использовать RAG для улучшения NLP систем
-
-

Знакомство с аудиторией



Кратко представьтесь и ответьте на следующие вопросы:

1. В какой области вы обучаетесь/работаете?
2. Слышали ли вы о RAG?

Теория

IBM: What is Retrieval-Augmented Generation (RAG)?

<https://www.youtube.com/watch?v=T-D1OfcDW1M>

Что такое RAG?

- **Retrieval-Augmented Generation (RAG)** - это
 - МЕТОД оптимизации ответа большой языковой модели (**LLM**),
 - КОТОРЫЙ комбинирует генерацию текста с поиском информации в базе знаний
 - Для создания более информативных ответов или текстов.

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;
plexis@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 systems that can retrieve facts. Additionally, precisely predicting where to store knowledge and updating their world knowledge remains an open research problem. 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 to generate responses. Our proposed model, REALM, uses a seq2seq model and the non-parametric memory is a pre-trained seq2vec model and the 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 retrieval-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 [17]. They can do so without any explicit external memory, such as an 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,

Источник: <https://arxiv.org/pdf/2005.11401>



Зачем нужен RAG в NLP сервисах?

- **Вызовы в работе с LLM:**
 - Могут создавать вводящие в заблуждение "**галлюцинации**"
 - Полагаются на потенциально **устаревшую информацию**
 - Неэффективны при работе с **конкретными знаниями**
 - Не имеют глубины в **специализированных областях**
 - Недостаточно развиты в области **логических способностей**
 - Лишены **контролируемости**
 - Не могут отслеживать **источник знаний**
 - Не могут обеспечить **конфиденциальность данных**
 - **Стоимость** их обучения высока
- **RAG улучшает точность и соответствие** результатов за счёт использования контекстуально значимой информации из базы данных
- При отсутствии информации может ответить "я не знаю" вместо убедительного, но вымышленного или неправильного ответа



Зачем нужен RAG в NLP сервисах?

Вопрос: «У какой планеты в солнечной системе больше всего спутников?» (ChatGPT vs NASA)

AN Du

which planet has the most moons in the solar system?

ChatGPT

Jupiter holds the title for having the most moons in our solar system. As of my last update, it boasts a whopping 79 moons confirmed by NASA. Ganymede, one of Jupiter's moons, is the largest moon in the solar system, even bigger than the planet Mercury.



ChatGPT (gpt-3.5-turbo), 12.05.2024

<https://platform.openai.com/docs/models/gpt-3-5-turbo>

According to the NASA/JPL Solar System Dynamics team, the current tally of moons orbiting planets in our solar system is 293: One moon for Earth; two for Mars; 95 at Jupiter; 146 at Saturn; 28 at Uranus; 16 at Neptune; and five for dwarf planet Pluto.

NASA, 12.05.2024

<https://science.nasa.gov/solar-system/moons/>

MODEL	DESCRIPTION	CONTEXT WINDOW	TRAINING DATA
gpt-3.5-turbo-0125	New Updated GPT 3.5 Turbo The latest GPT-3.5 Turbo model with higher accuracy at responding in requested formats and a fix for a bug which caused a text encoding issue for non-English language function calls. Returns a maximum of 4,096 output tokens. Learn more.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo	Currently points to gpt-3.5-turbo- ----	16,385 tokens	Up to Sep 2021

Сервис, использующий RAG, мог бы дать правильный ответ автоматически



Области применения RAG

- **Вопросно-ответные системы:**
 - Позволяет модели лучше понимать контекст запроса и формировать более информативные ответы.
- **Суммаризация текста:**
 - Позволяет интегрировать контекст из базы знаний для создания более полных и точных сводок
- **Другие области:**
 - Везде, где требуется создание текста на основе доступной информации из базы знаний



Типы RAG

1. Классический RAG (для QA) включает три основных шага:

1. **Индексация (indexing)** - разделение корпуса документов на более короткие фрагменты и построение векторного индекса с помощью энкодера
2. **Извлечение (retrieval)** - извлечение соответствующих фрагментов документов на основе сходства между вопросом и фрагментами
3. **Генерация (generation)** - генерация ответа на вопрос с учетом извлеченного контекста

Типы RAG

2. Продвинутый RAG

Продвинутая парадигма RAG включает **дополнительную обработку перед извлечением и после извлечения:**

1. Перед извлечением методы, такие как переформулировка запроса, могут использоваться для выравнивания семантических различий между вопросами и фрагментами документов.
2. После извлечения, ранжирование корпуса документов может помочь избежать явления "потерянный в середине"*, или контекст может быть отфильтрован и сжат, чтобы сократить длину контекстного окна.

* Как и люди, LLM с большей вероятностью вспомнят информацию, расположенную в начале или в конце документа, при этом они склонны игнорировать контент в середине.

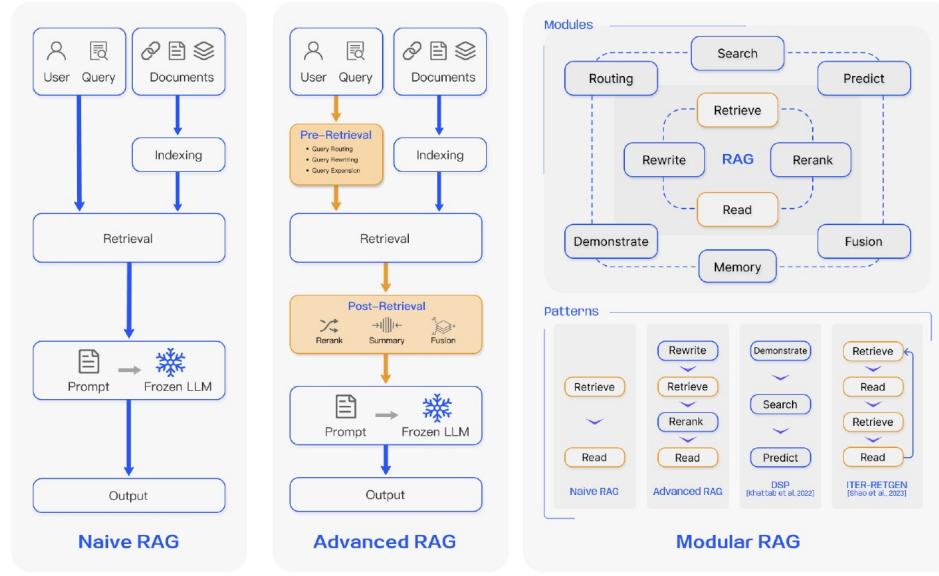
Согласно результатам исследований, производительность LLM оптимальна, когда соответствующие данные расположены в начале или в конце документа.



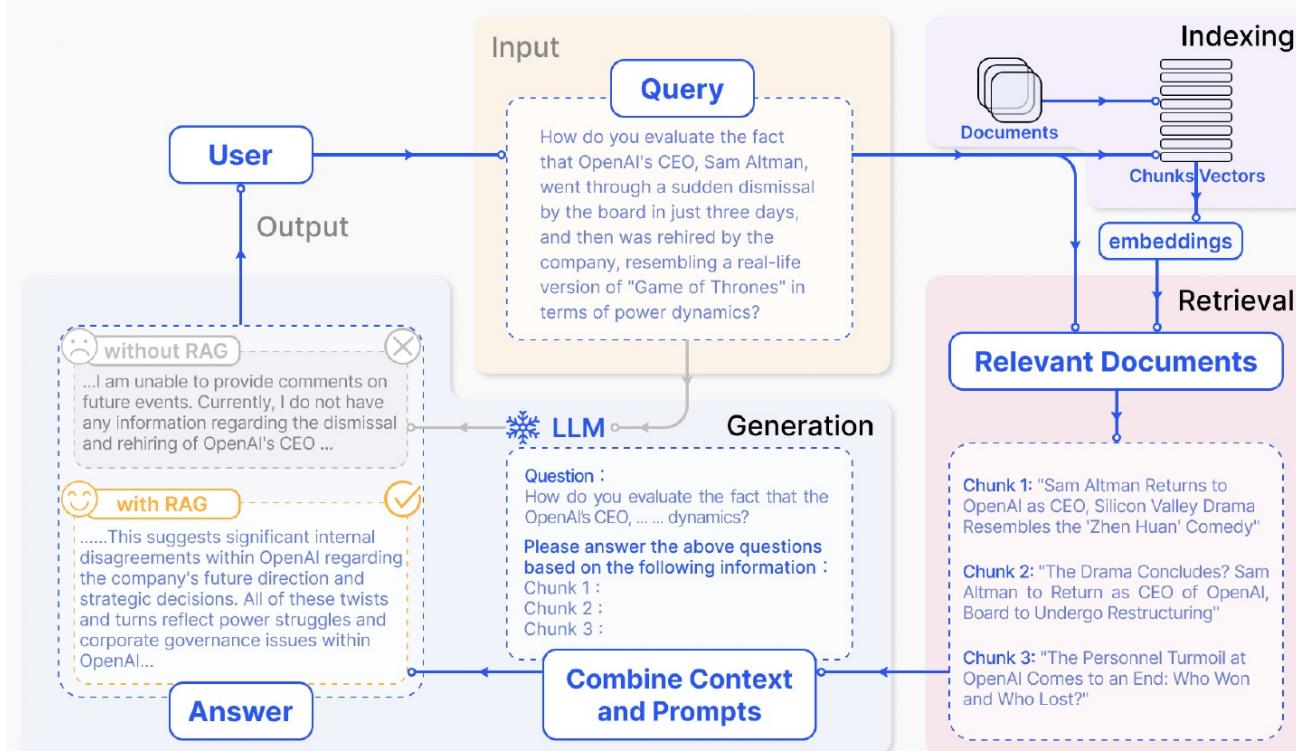
Типы RAG

3. Модульный RAG:

- Структурно он более свободен и гибок, в нем представлены более специфические функциональные модули, такие как **поисковые системы** запросов и **объединение нескольких ответов**
- Технологически он **объединяет поиск с fine-tuning, обучением с подкреплением** и другими методами



Пример архитектуры сервиса с RAG



RAG vs Fine-Tuning

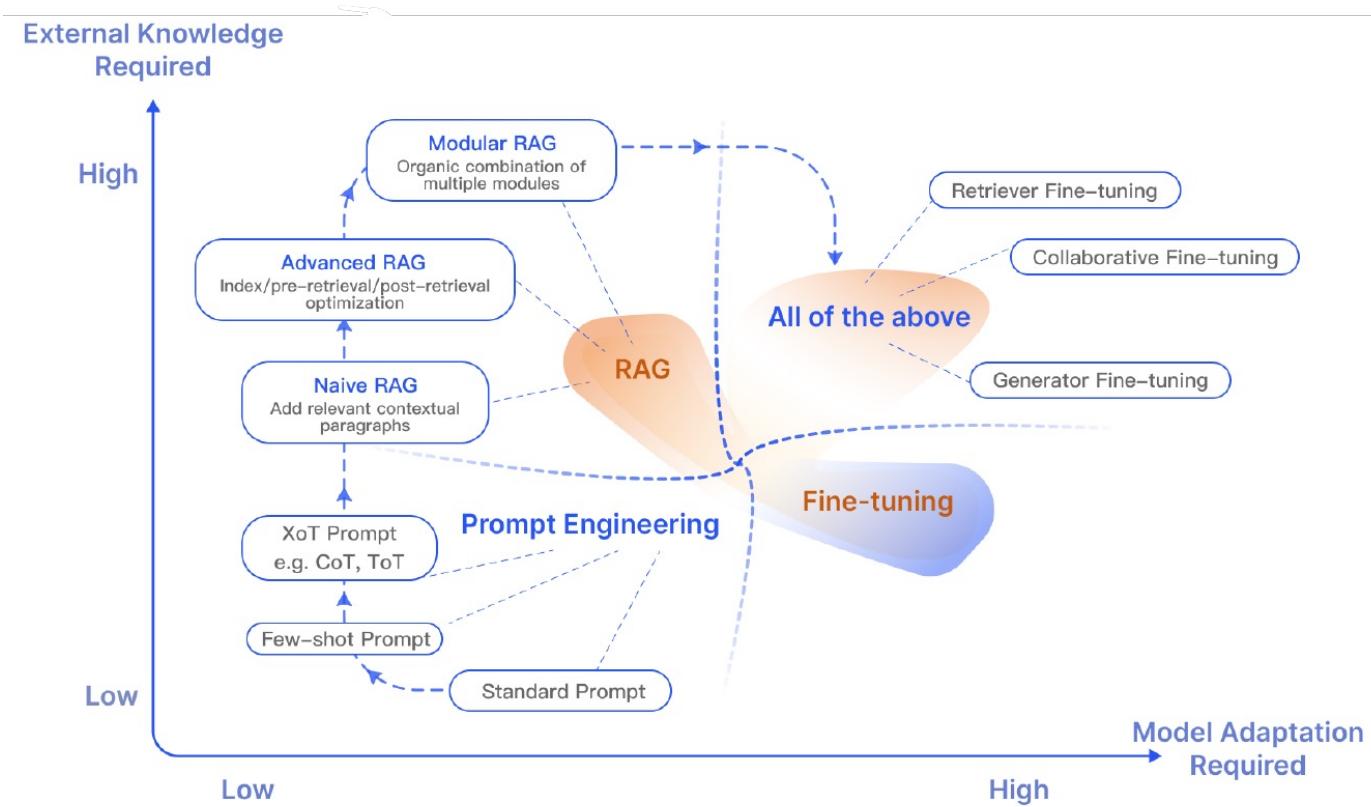
- У обоих методов есть преимущества и недостатки
- **Наилучшие результаты можно получить при совместном использовании RAG, Fine-tuning и Prompt Engineering**

Table 1: Comparison between RAG and Fine-Tuning

Feature Comparison	RAG	Fine-Tuning
Knowledge Updates	Directly updating the retrieval knowledge base ensures that the information remains current without the need for frequent retraining, making it well-suited for dynamic data environments.	Stores static data, requiring retraining for knowledge and data updates.
External Knowledge	Proficient in leveraging external resources, particularly suitable for accessing documents or other structured/unstructured databases.	Can be utilized to align the externally acquired knowledge from pretraining with large language models, but may be less practical for frequently changing data sources.
Data Processing	Involves minimal data processing and handling.	Depends on the creation of high-quality datasets, and limited datasets may not result in significant performance improvements.
Model Customization	Focuses on information retrieval and integrating external knowledge but may not fully customize model behavior or writing style.	Allows adjustments of LLM behavior, writing style, or specific domain knowledge based on specific tones or terms.
Interpretability	Responses can be traced back to specific data sources, providing higher interpretability and traceability.	Similar to a black box, it is not always clear why the model reacts a certain way, resulting in relatively lower interpretability.
Computational Resources	Depends on computational resources to support retrieval strategies and technologies related to databases. Additionally, it requires the maintenance of external data source integration and updates.	The preparation and curation of high-quality training datasets, defining fine-tuning objectives, and providing corresponding computational resources are necessary.
Latency Requirements	Involves data retrieval, which may lead to higher latency.	LLM after fine-tuning can respond without retrieval, resulting in lower latency.
Reducing Hallucinations	Inherently less prone to hallucinations as each answer is grounded in retrieved evidence.	Can help reduce hallucinations by training the model based on specific domain data but may still exhibit hallucinations when faced with unfamiliar input.
Ethical and Privacy Issues	Ethical and privacy concerns arise from the storage and retrieval of text from external databases.	Ethical and privacy concerns may arise due to sensitive content in the training data.



RAG vs Fine-Tuning



Методы и инструменты для оценки системы RAG

Table 2: Summary of metrics applicable for evaluation aspects of RAG

	Context Relevance	Faithfulness	Answer Relevance	Noise Robustness	Negative Rejection	Information Integration	Counterfactual Robustness
Accuracy	✓	✓	✓	✓	✓	✓	✓
EM					✓		
Recall	✓						
Precision	✓			✓			
R-Rate							✓
Cosine Similarity			✓				
Hit Rate	✓						
MRR	✓						
NDCG	✓						

Table 3: Summary of evaluation frameworks

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

- 3 качества:
 - релевантность контекста
 - достоверность ответа
 - релевантность ответа
- Оценка включает 4 ключевые возможности:
 - устойчивость к помехам в данных (noise),
 - способность к отказу (I don't know)
 - интеграция информации
 - устойчивость к фейкам
- Системы оценки (бенчмарки)
 - RGB
(Retrieval-Augmented Generation Benchmark)
 - RECALL
- автоматизированные инструменты оценки:
 - RAGAS
 - ARES
 - TruLens



Мониторинг RAG в production

- Инструменты:
 - Langsmith
 - Phoenix (Arize)
 - Langfuse
 - OpenLayer
- Аспекты мониторинга :
 - **Достоверность** - помогает выявлять и количественно оценивать случаи галлюцинаций
 - **Плохое извлечение** - помогает выявить и количественно оценить плохое извлечение контекста
 - **Плохой ответ** - помогает распознать и количественно оценить уклончивые, вредные или токсичные ответы
 - **Плохой формат** - позволяет обнаружить и количественно оценить ответы с неправильным форматированием



Текущие проблемы в RAG:

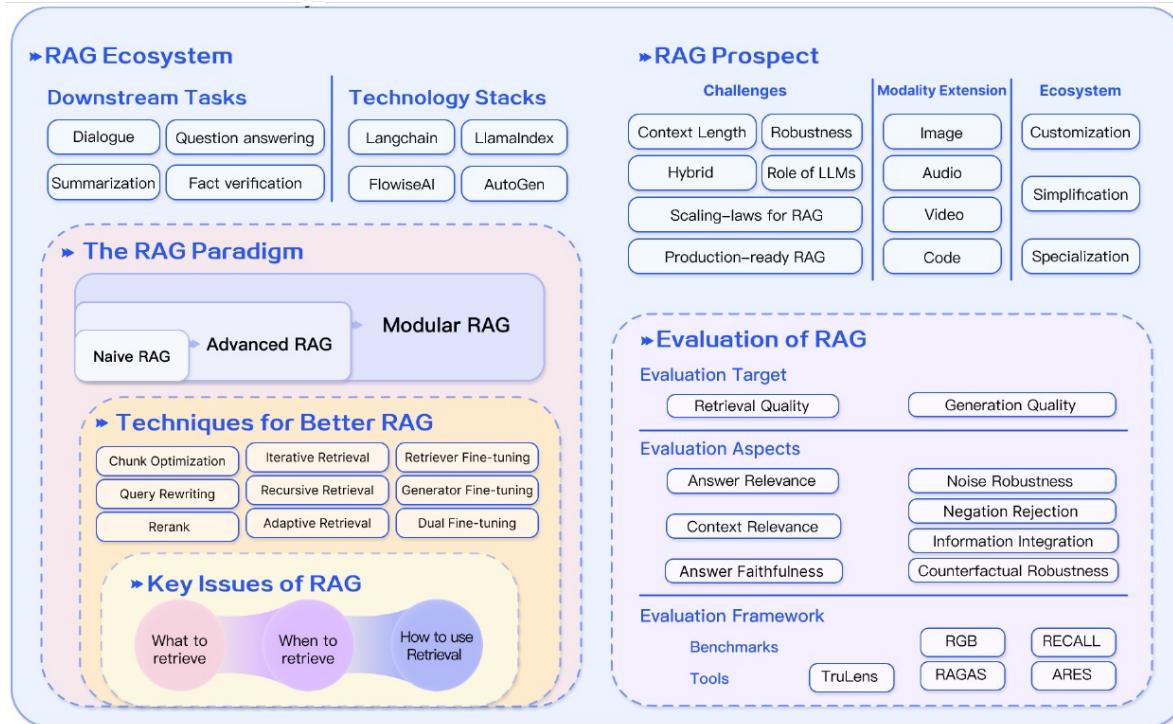
- **Длина контекста:**
 - Что делать, если полученный **контент слишком велик** и превышает лимит контекстного окна?
 - Если контекстное окно LLM больше не ограничено, как улучшить RAG?
- **Надежность:**
 - Как **бороться с некорректным** полученным **контентом**?
 - Как **фильтровать** и проверять полученный **контент**?
 - Как повысить **устойчивость** модели **к засорениям и шуму**?
- **Координация с fine-tuning:**
 - Как использовать эффекты RAG и FT одновременно?
 - Как они должны **координироваться**, организовываться - последовательно, попаременно или e2e?
- **Законы масштабирования:**
 - Удовлетворяет ли модель RAG закону масштабирования?
 - При каких сценариях RAG может столкнуться с явлением обратного закона масштабирования (**падение производительности**)?



Текущие проблемы в RAG:

- **Роль LLM:**
 - LLM могут быть использованы для поиска, для генерации, для оценки.
 - Как дальше **исследовать потенциал LLM** в RAG?
- **Готовность к prod.:**
 - Как уменьшить **время поиска** в сверхбольших базах данных?
 - Как гарантировать, что **найденный контент не утечет из LLM**?
- **Мультимодальное расширение:**
 - Как можно распространить развивающиеся технологии и концепции RAG на другие виды данных, такие как изображения, аудио, видео или код?

«RAG-вселенная»:



Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Демонстрация QA сервиса с RAG

Huggingface RAG:

https://github.com/huggingface/transformers/tree/main/examples/research_projects/rag

Huggingface RAG end2end retriever:

https://github.com/huggingface/transformers/tree/main/examples/research_projects/rag-end2end-retriever

LLaVA vison model:

<https://llava-vl.github.io>

Langchain chain types:

<https://python.langchain.com/v0.1/docs/modules/chains/>

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Рефлексия

Цели вебинара

Проверка достижения целей

1. Объяснить подход RAG
 2. Использовать RAG для улучшения NLP систем
-

Рефлексия



С какими впечатлениями уходите с вебинара?



Хотите дальше разбираться с RAG и с NLP в целом?