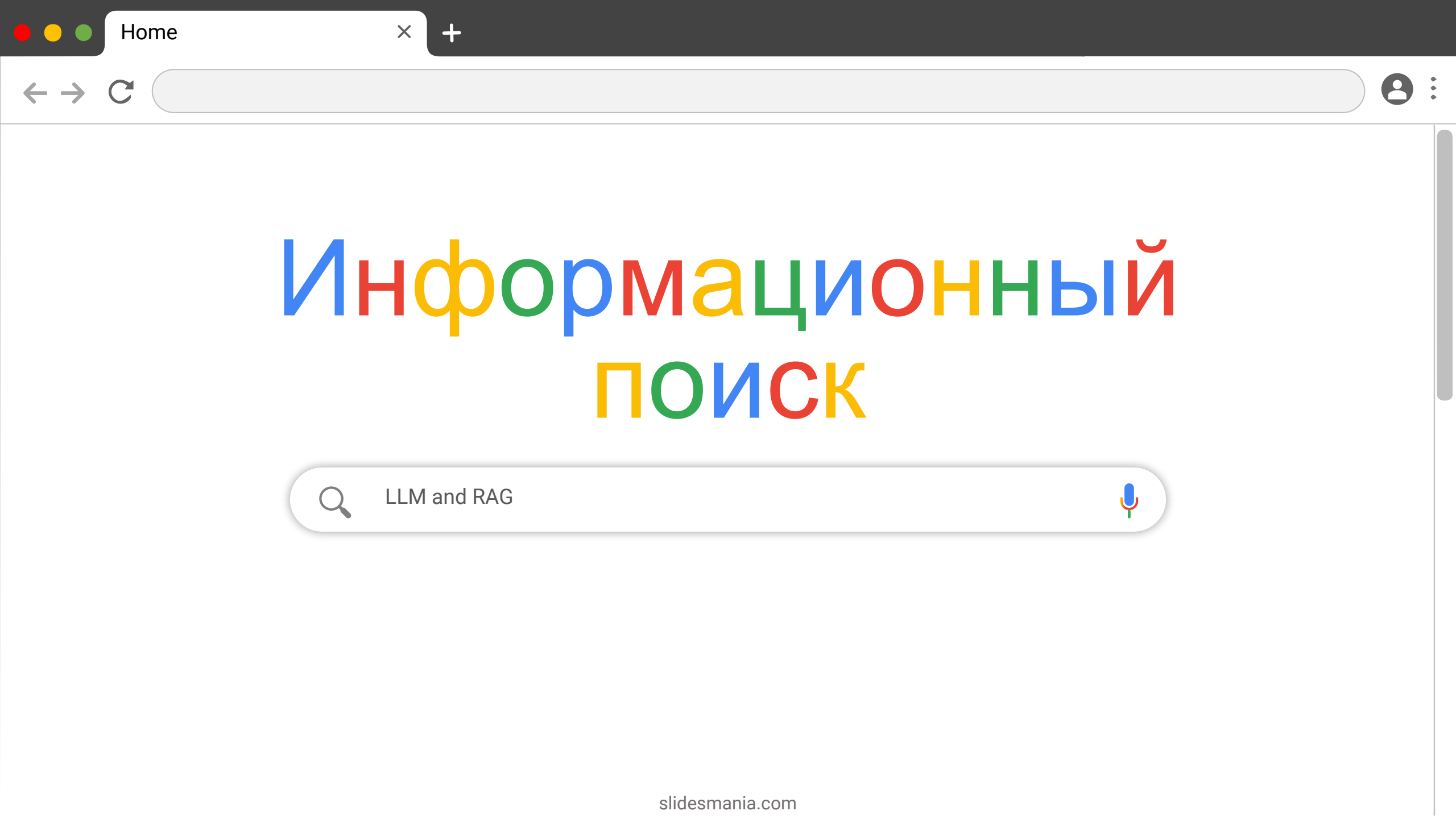

Home
Информационный поиск
LLM and RAG
slidesmania.com

Large Language Models

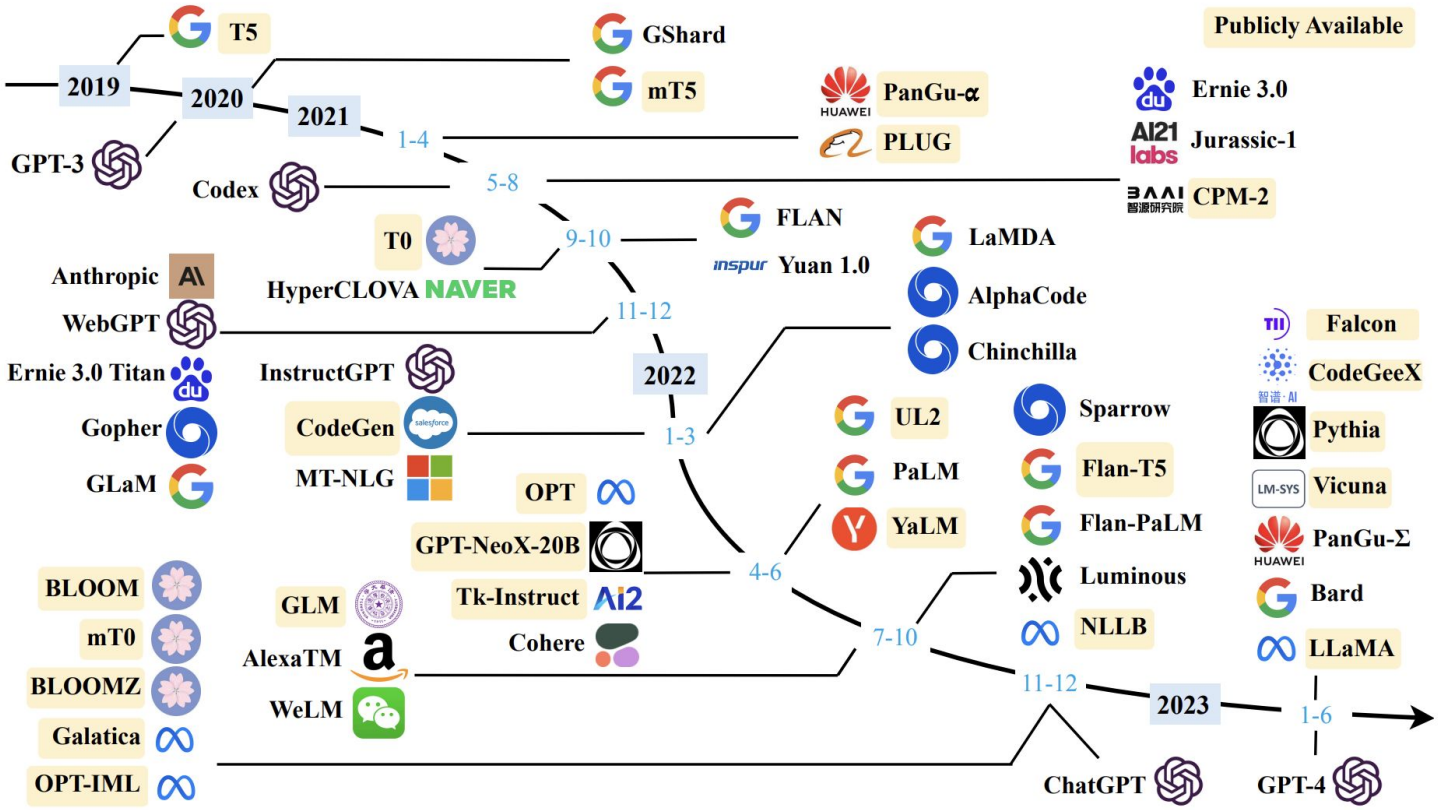
На данный момент большие языковые модели являются State-of-the-Art решениями во многих областях NLP (и не только).

При этом вокруг них активно выстраивается целая экосистема: API, специальные библиотеки, приемы использования и дообучения на малых мощностях.

Мы сегодня поговорим про LLM в целом, а также про приемы промптинга и RAG.

Источник графики: https://yellow.ai/wp-content/uploads/2023/11/LLM--1024x1024.webp

Large Language Models

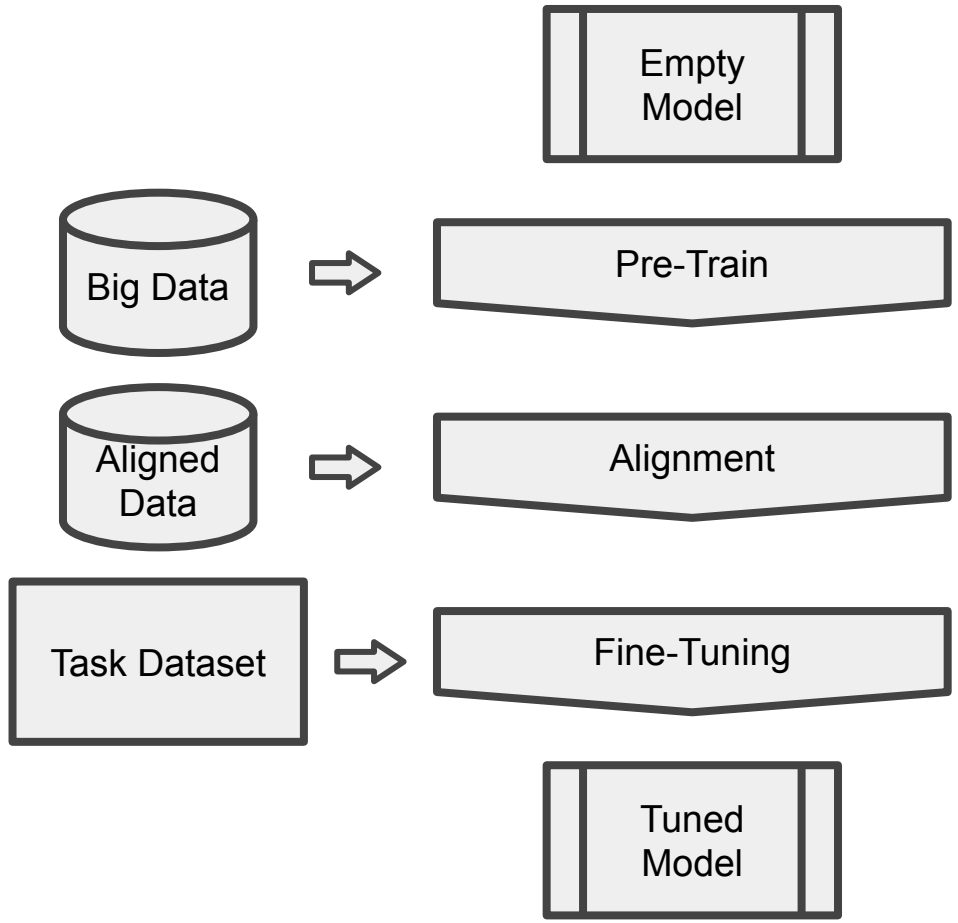


Источник графики: https://arxiv.org/pdf/2303.18223.pdf

Процесс обучения

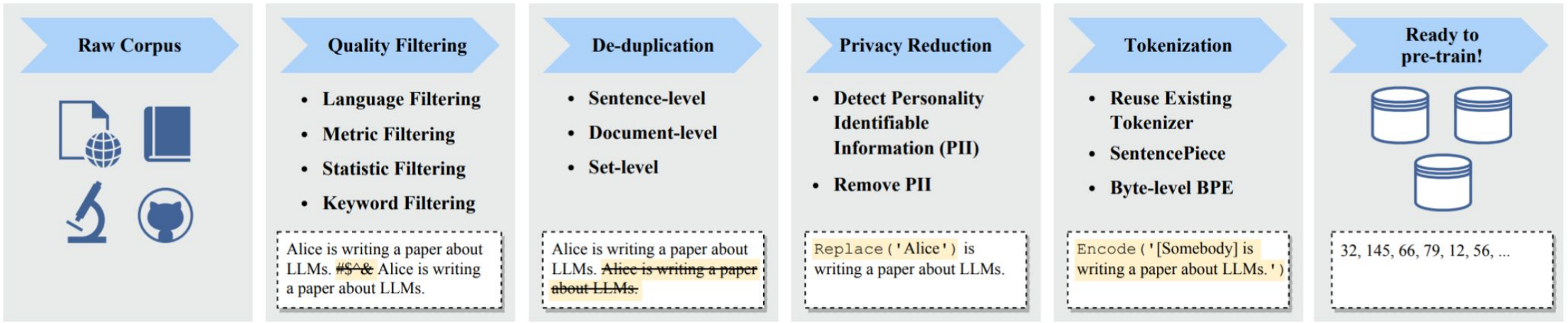
Обычно процесс разделяют на три больших этапа:

1. **Pre-train:** на данном этапе происходит основное обучение, о котором мы говорили выше. Длинный и вычислительно дорогой процесс
2. **Alignment** (обычно выделяют для llm): здесь модель учится соблюдать определенный набор правил, “выравнивается” под некоторое распределение
3. **Fine-tuning:** дообучение модели на конкретную задачу



Процесс обучения: pre-train

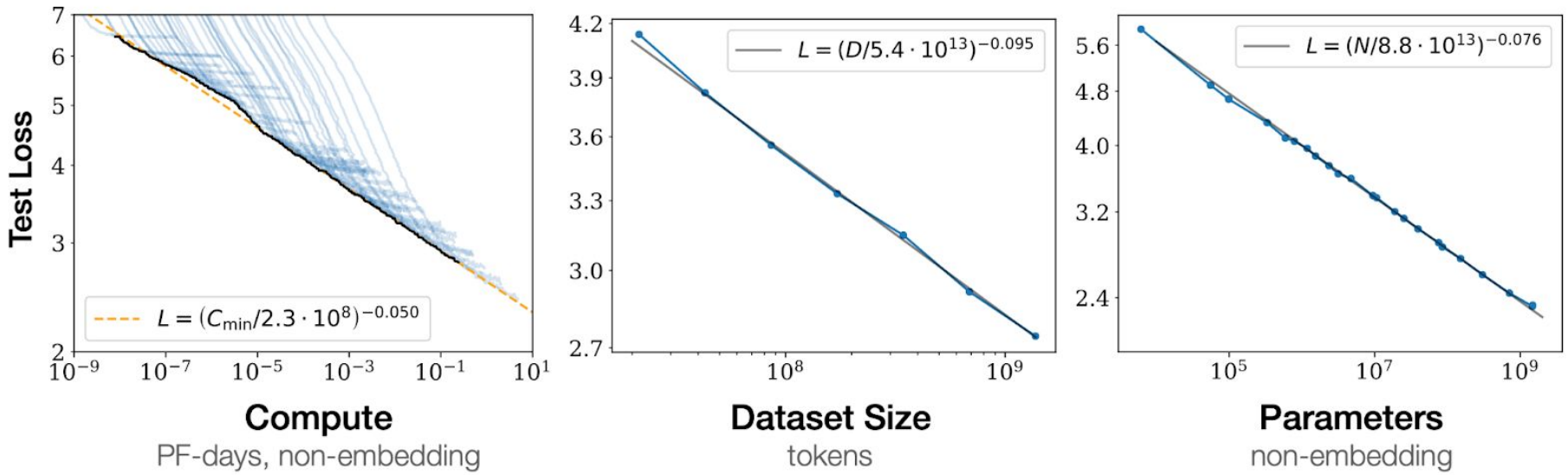
1. Обычно говорят, что такие модели обучаются на сырых данных, однако препроцессинг все-таки нужен:
2. Бывают дубли в данных, которые увеличивают время обучения, но не дают информации
3. Персональные данные лучше не включать в такие модели
4. Предложения на других языках тоже



Источник изображения: https://arxiv.org/pdf/2303.18223.pdf

Параметры обучения

1. Обучение модели обычно существенно зависит от трех параметров: количества итераций обучения, размера датасета и размера самой модели.
2. И, конечно, архитектуры, но тут чаще всего выбор между Трансформером и Тра



Источник изображения: https://sysblok.ru/knowhow/kak-rabotajut-jazykovye-modeli/

Вот как-то так

Home | Recent Tabs | Current Page

You can type something here...

Процесс обучения: alignment

На данный момент есть целая активно развивающаяся область на стыке AI, информационной безопасности и этики, которая занимается вопросами элаймента.

Безопасность и этика

Учим модель правильно реагировать на запросы:

- Никакой симметричной реакции на негатив
- Нельзя разглашать персональные данные

Реакция на инструкции

Современная llm должна уметь следовать произвольным набором инструкций:

- “Напиши код для задачи”
- “Расскажи рецепт тирамису”

Выбросы и смещения

Модель не должна галлюцинировать и выходить из строя на странных примерах:

- Мир меняется
- Пользователи не всегда пишут что-то осмысленное.

Источник изображения: https://sysblok.ru/knowhow/kak-rabotajut-jazykovye-modeli/

Процесс обучения: fine-tuning

Существует множество задач, на которые можно дообучать модель.

Encoder + Classification Head

- Natural Language Inference
- Sentiment Analysis
- Spam Detection
- Hate Speech Recognition
- Topic classification

Decoder | Encoder-Decoder

- Summarization
- Question Answering
- Machine Translation
- Style Transfer
- Text Generation in general

Источник изображения: https://sysblok.ru/knowhow/kak-rabotajut-jazykovye-modeli/

Промптинг

Если модель была дообучена на следование инструкциям, то ее называют instruction-tuned (чаще всего обозначается -it в названии модели). Такие модели лучше всего реагируют на промпты и письменные инструкции, так что для всего, что будет обсуждаться дальше, лучше использовать их.

Промптинг - это процесс подбора затравки (промпта) для модели так, чтобы результат генерации лучше всего соответствовал ожиданиям. Он бывает ручной, когда мы просто придумываем промпт, и автоматический, куда относятся техники RAG, обучение моделей-промптеров и прочие похожие подходы.

Источник изображения: https://sysblok.ru/knowhow/kak-rabotajut-jazykovye-modeli/

Home | Recent Tabs | Current Page

You can type something here...

Промптинг: виды

Самые частые виды промптинга:

- Zero-shot
- Few-shot
- Chain-of-Thought (CoT)
- Self-Consistency
- Retrieval Augmented Generation (RAG)

Prompt:

```
Classify the text into neutral, negative or positive.
Text: I think the vacation is okay.
Sentiment:
```

Output:

```
Neutral
```

Prompt:

```
This is awesome! // Negative
This is bad! // Positive
Wow that movie was rad! // Positive
What a horrible show! //
```

Output:

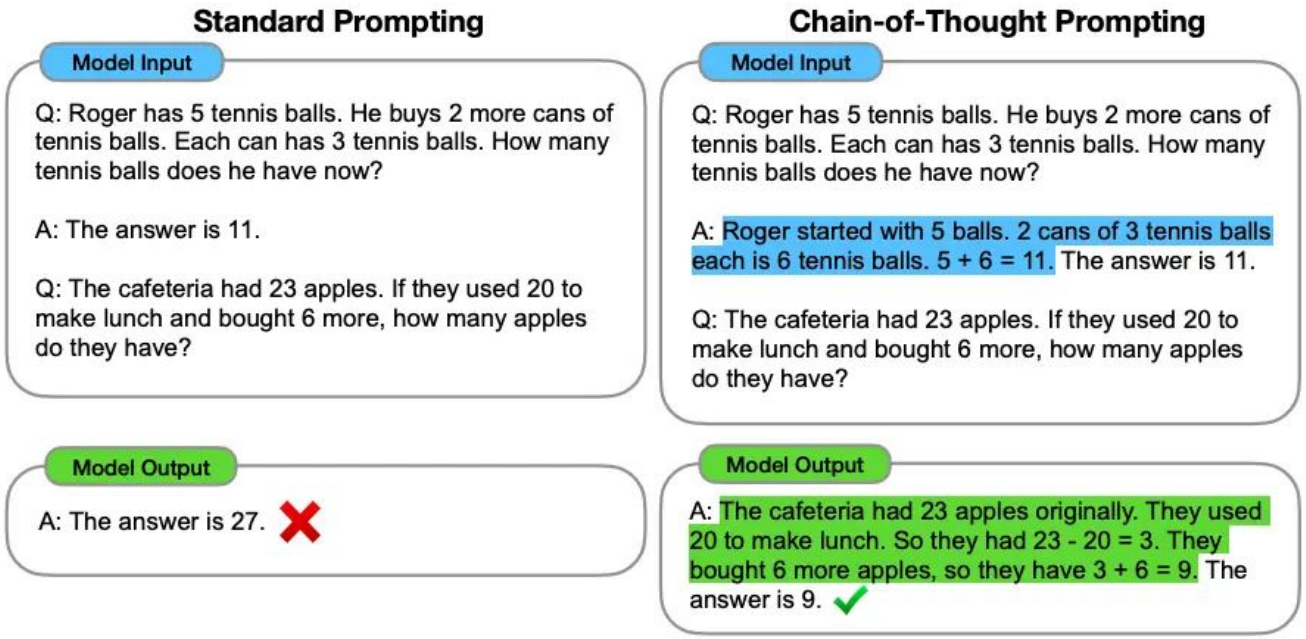
```
Negative
```

Источник изображения: https://www.promptingguide.ai/introduction/basics

Промптинг: Chain-of-Thought

Вместо того, чтобы сразу требовать ответ на поставленный вопрос, мы даем модели пример рассуждений и просим от нее думать похожим образом.

На практике такой метод оказывается дороже (почему?), но эффективнее, особенно в математических задачах.



Источник изображения: https://www.promptingguide.ai/introduction/basics

Промптинг: Self-Consistency

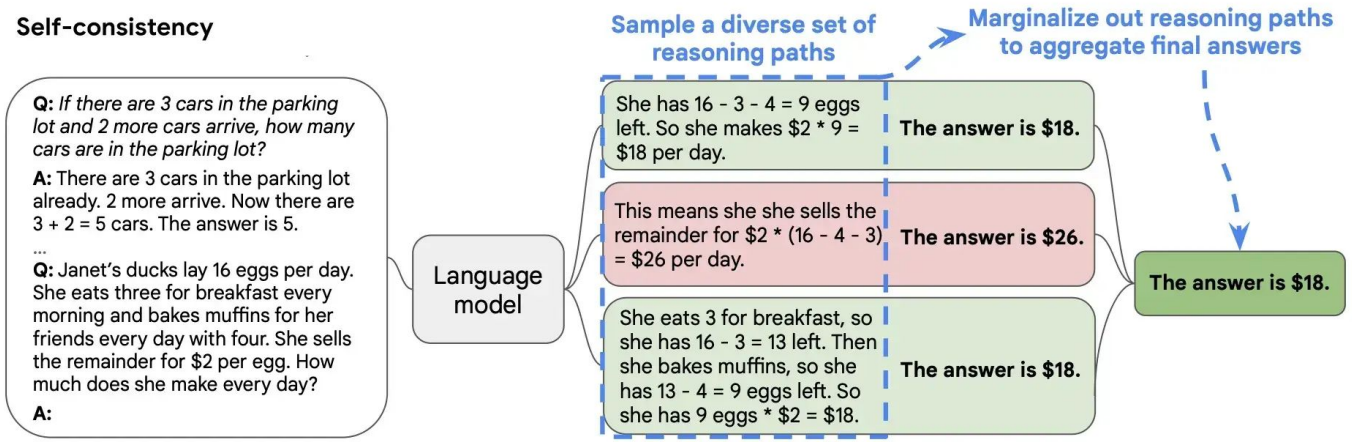
Похоже на цепочки рассуждения, но здесь мы генерируем несколько вариантов таких цепочек и потом берем средний (или самый частотный) результат их полученных.

Плюсы:

- Можем избежать принятия галлюцинации как правильного ответа, так как она менее вероятно повторится

Минусы:

- Еще более дор



Источник изображения:
https://learnprompting.org/ru/docs/intermediate/self_consistency?srsltid=AfmBOooU8AfGVzYIHqf7jrO_oBfo4Oet5qVihsmkWfEkJbB3irYUINgv

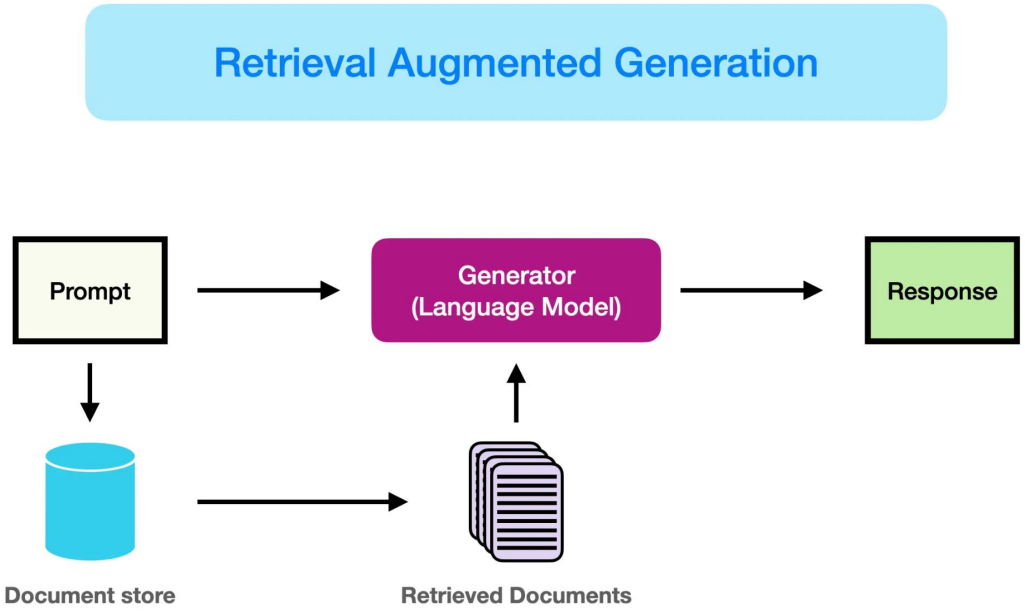
Промптинг: Retrieval Augmented Generation

Здесь мы улучшаем генерацию с помощью стороннего источника знаний.

Как это работает:

- Векторизуем все документы в нашей базе
- Когда нам поступаем промпт, ищем топ-n ближайших документов
- На вход LLM подаем промпт плюс найденные документы

Какие здесь есть трудности?

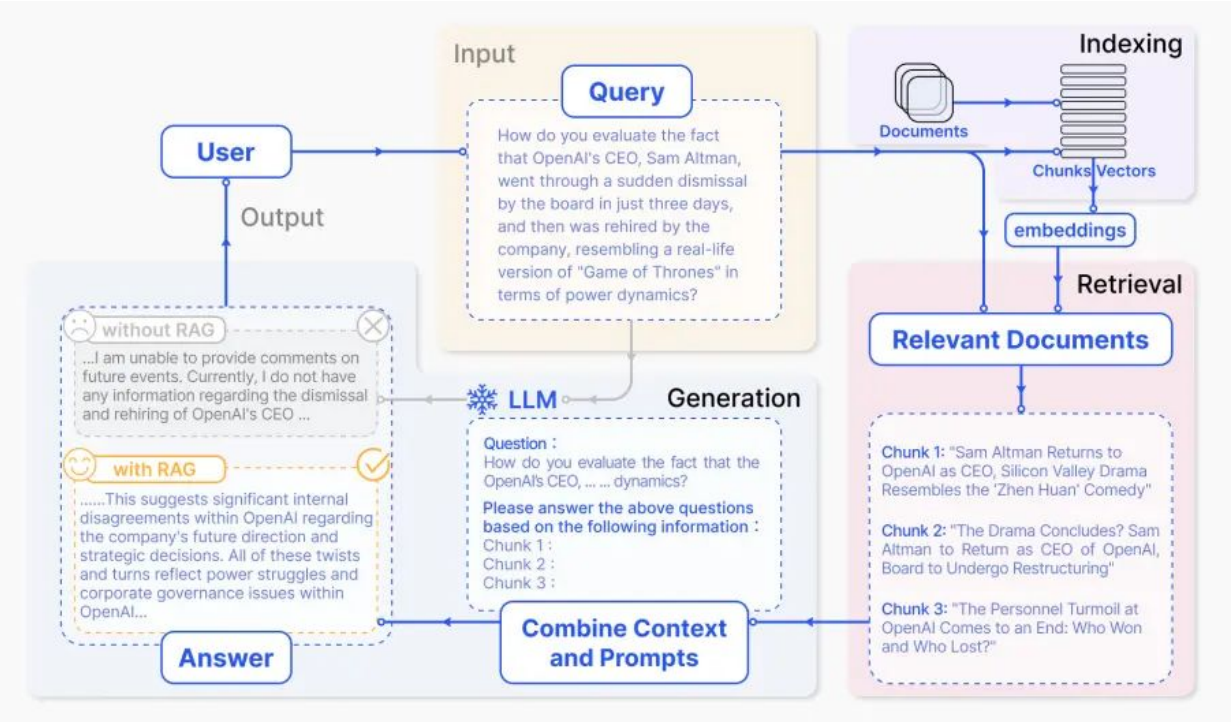


Источник изображения: https://www.promptingguide.ai/research/rag

Промптинг: Retrieval Augmented Generation

Хитрости:

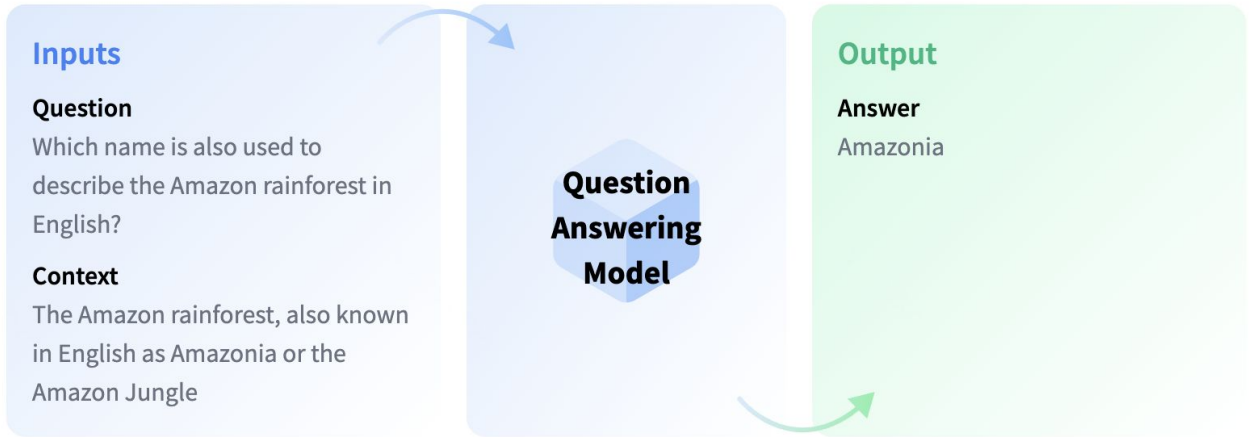
- Можно делить тексты на чанки так, чтобы не разрывать информацию, но при этом сократить размер промпта
- Если использовать для векторизации саму LLM, то можно попробовать пропустить какое-то кол-во вычислений (почему это на практике сложно реализовать?)
- Иногда можно давать не все параграфы, а заголовки/результаты суммаризации



Источник изображения: https://www.promptingguide.ai/research/rag

Если нет LLM

В чем-то все описанное похоже на задачу QA, просто используется большая и тяжелая модель, которую не надо дообучать. Но если у вас есть время и данные, но нет LLM, то можно обучить генеративную модель вроде T5, чтобы она генерировала вам ответы по нескольким абзацам, которые вы будете подбирать по логике RAG.



Источник изображения: https://huggingface.co/tasks/question-answering