Yandex Cloud

Построение RAG-приложения с использованием YaGPT

Кузьма Лешаков
Yandex Cloud

Кузьма Лешаков

≈ 8 лет в IT

2023 — Более года в Yandex Cloud
Команда Data Platform

2021 — Uma.Tech.
Ведущий инженер данных

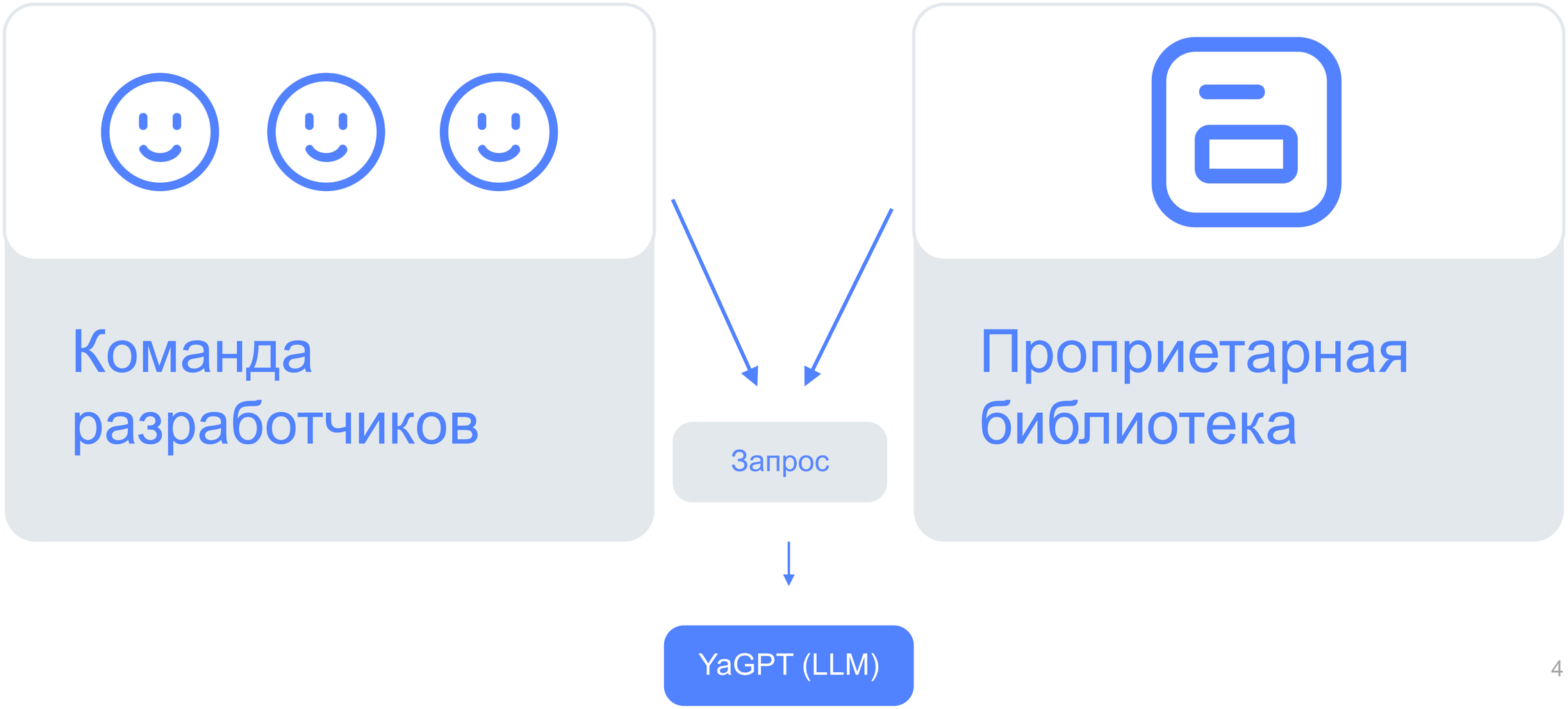
2019 — Clover Group
Инженер данных

2017 — The Linux Foundation
Стажёр-разработчик

О чём поговорим

1. Введение в YaGPT
2. Архитектура RAG приложения
3. Построение RAG приложения

Сценарий 1 – библиотека X



Языковые модели развиваются в Яндексе уже не первый год

2021

В июне — сервис «Зелибоба»/«Балабоба»

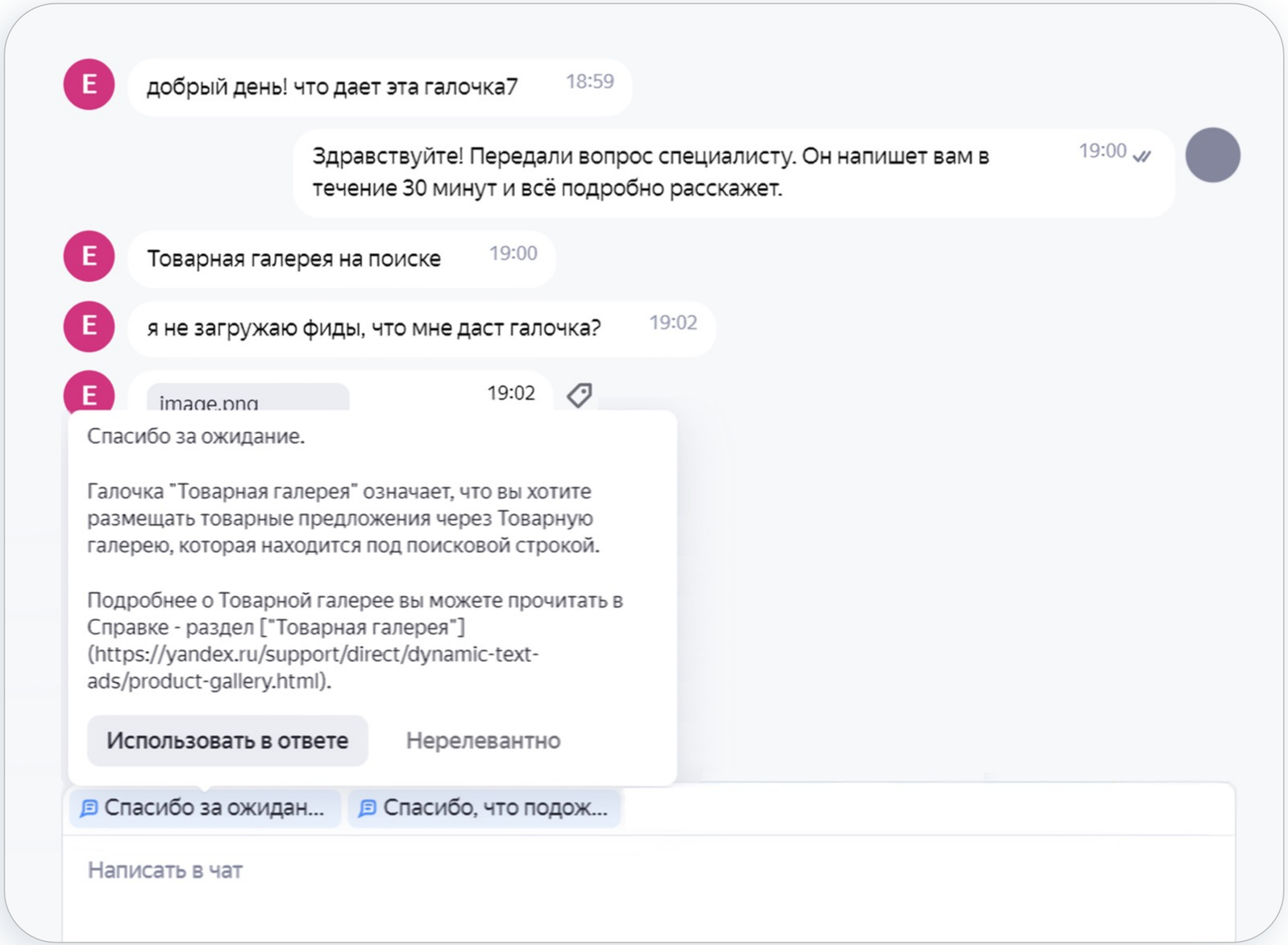
2022

В июне Яндекс выложил одну из крупнейших языковых моделей — YaLM 100B

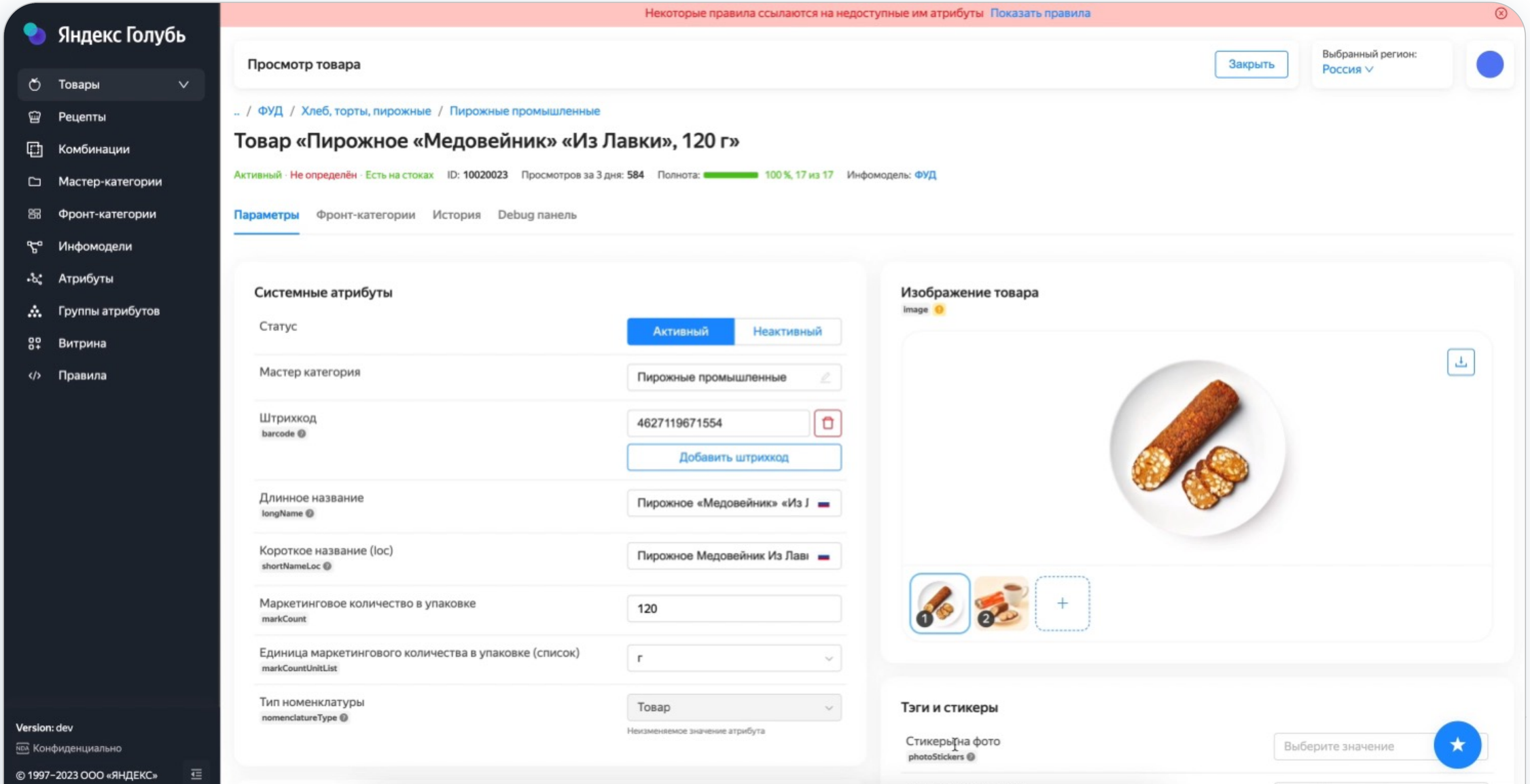
2023

Яндекс анонсировал выпуск новой модели YandexGPT

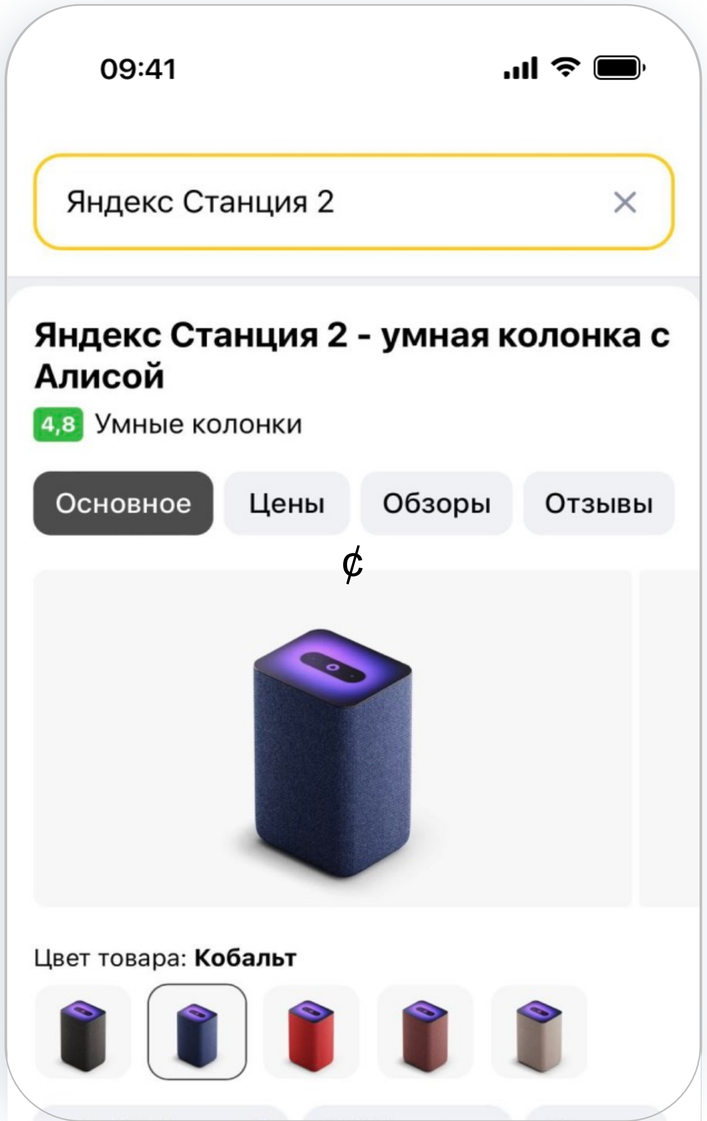
Лавка: Пример подсказок



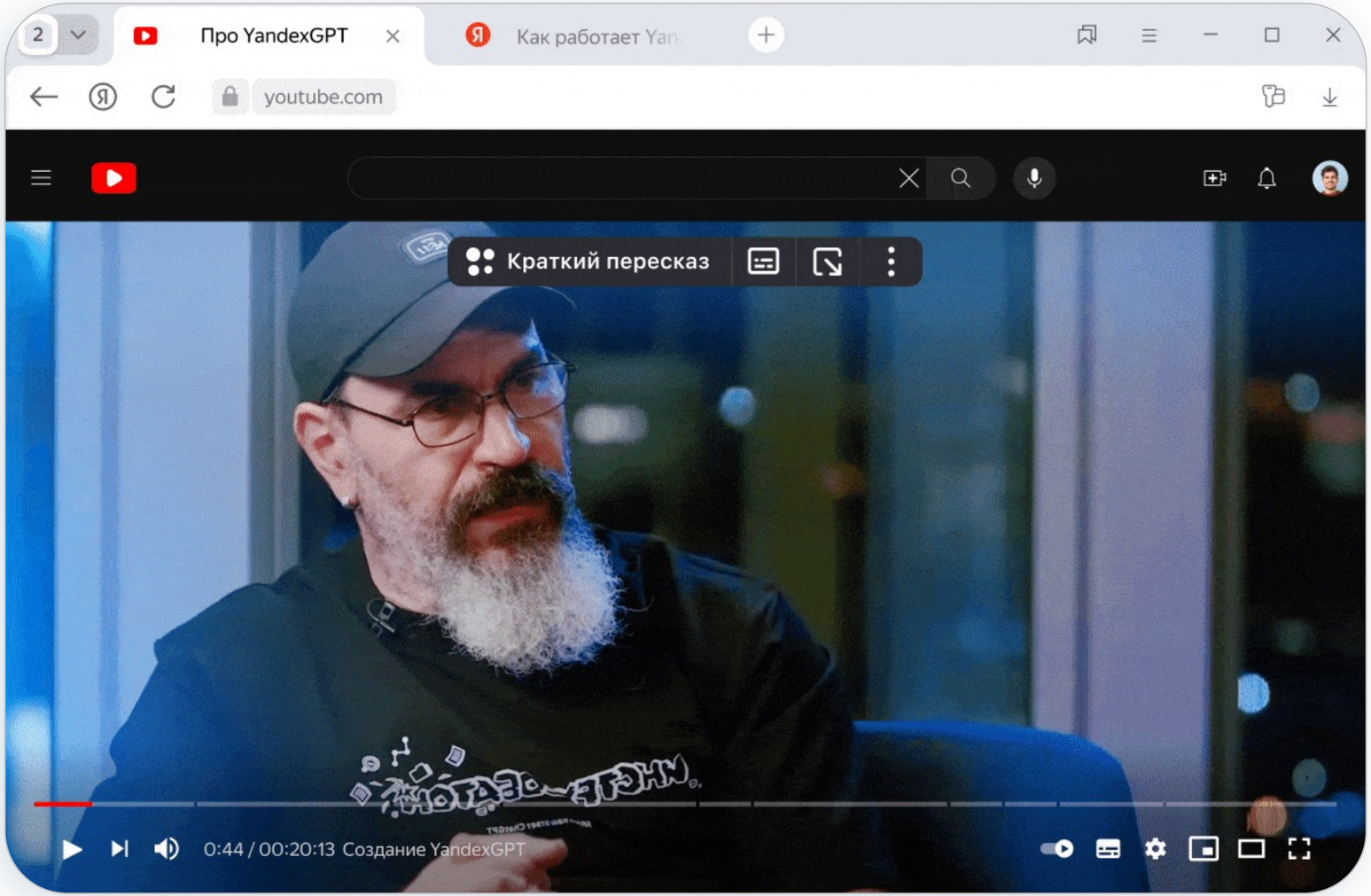
Лавка: генерация новых атрибутов



Суммаризация отзывов



Суммаризация видео



475 лет

сэкономлено
за месяц

LLM умеет: Хорошо vs Плохо

LLM умеет делать хорошо

- Генерировать и редактировать тексты
- Отвечать на вопросы по базе знаний
- Чатиться, общаться
- Обобщать и интерпретировать данные
- Классифицировать и определять тональность текста
- Выделять сущности из текста

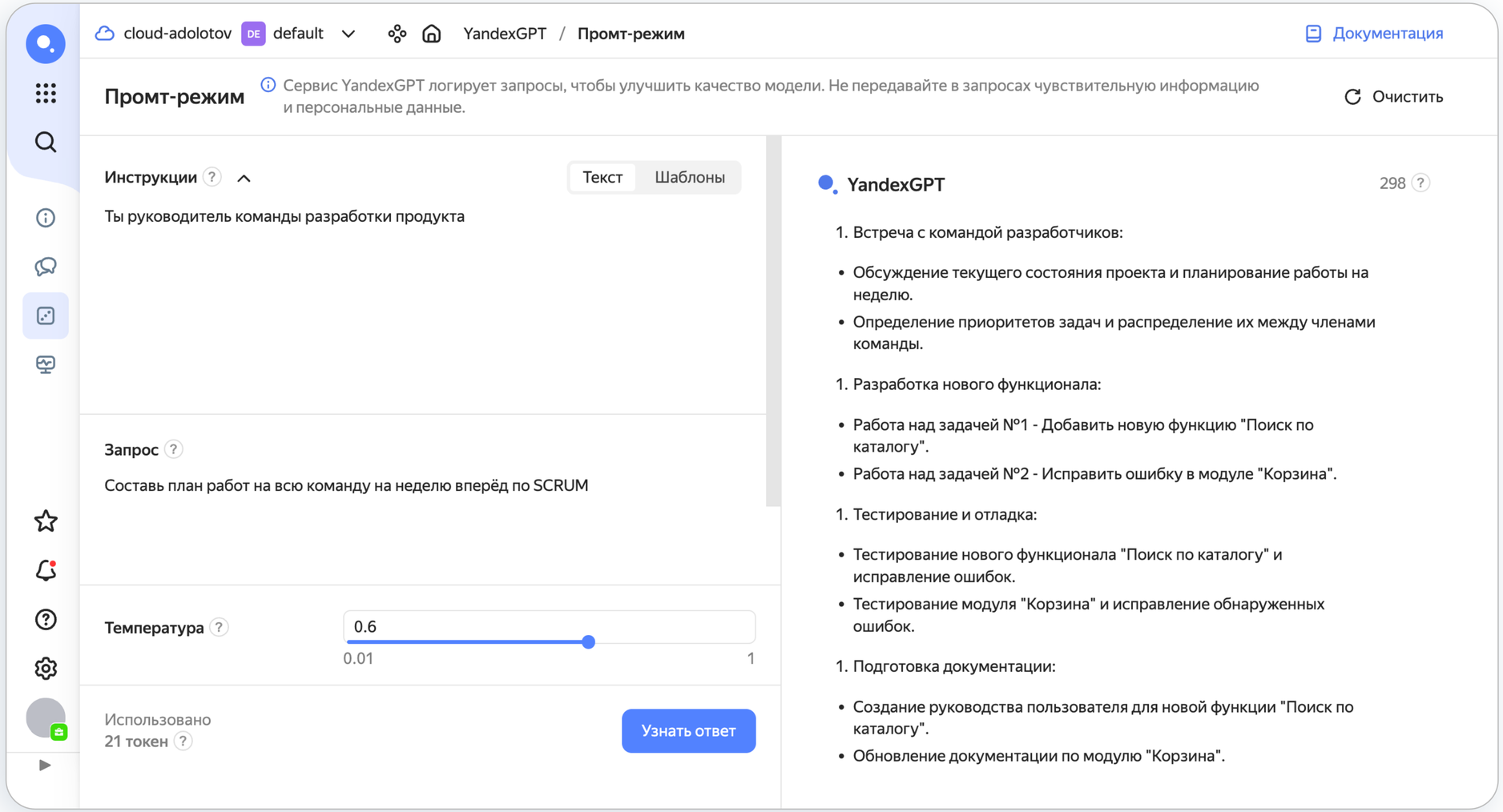
Есть решения, лучше LLM

- Ранжирование результатов поиска, рекомендации товаров, услуг и т. д.
- Решение задач, подразумевающих строгую логику
- Решение задач с низкой толерантностью к ошибке и высокой стоимостью проверки

Взаимодействие с YandexGPT API

Повышение качества моделей

Консоль: instruct режим



Обращение через API

```
{
 'modelUri': 'ds://<id_дообученной_модели>',
 'completionOptions':{
        'stream': False,
        'temperature': 0.1,
        'maxTokens': '1000'
 },
 'messages':[
       {
           'role': 'system',
           'text': '<текст вашей инструкции>'
       },
       {
           'role': 'user',
           'text': 'Теперь Яндекс Карты не только предупреждают о платных
                    участках на маршруте, но и показывают стоимость проезда.
                    Это позволяет водителям заранее оценить расходы
                    или выбрать маршрут без платных дорог, если это возможно...'
       }
    ]
}
```

YandexGPT API в Public Preview!

Доступные модели для всех пользователей

Модели

YandexGPT Lite

YandexGPT Pro

Эмбеддинги

Токенизатор

Суммаризация

На базе YaGPT Lite

Инференс

Синхронно

Стандарт

Асинхронно

Know-how

Повышение качества моделей

Какие есть способы работы с внешними источниками информации

Fine-tuning(*) модели

Для автоматического определения контекста / предметной области

Retrieval Augmented Generation (RAG)

Ответы по конкретным документам

RAG vs Fine-tuning(*) vs комбинация

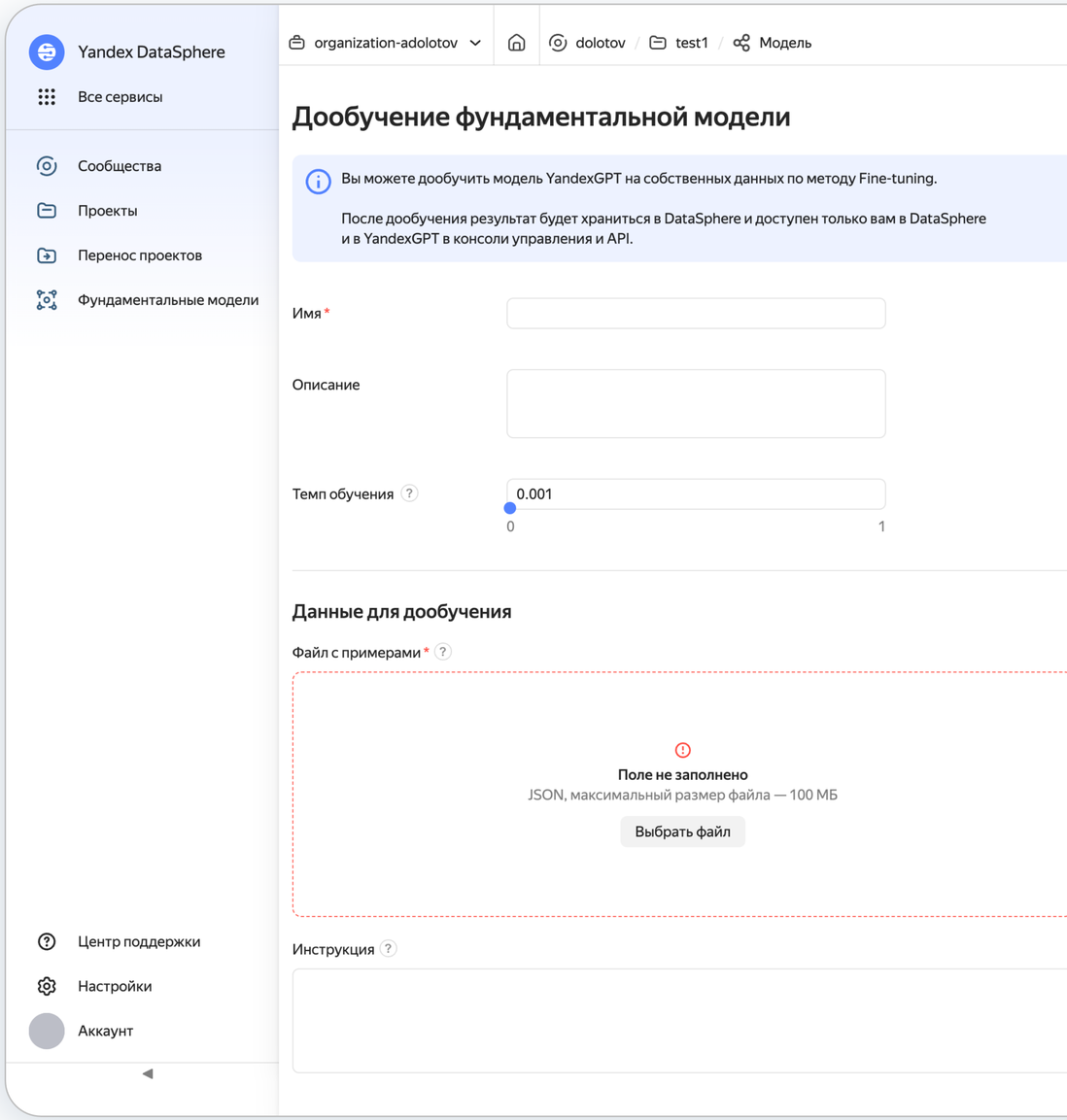
Аспект	RAG	P-tuning	Комбинация
Динамические данные	✓	✗	✓
Статические данные	✓	✓	✓
Внутренние данные	✓	✗	✓
Уменьшение галлюцинаций	✓	✓	✓
Прозрачность генерации	✓	✗	✓
Тонкая настройка под узкую задачу	✗	✓	✓
Голос бренда	✗	✓	✓

Реализация в DataSphere

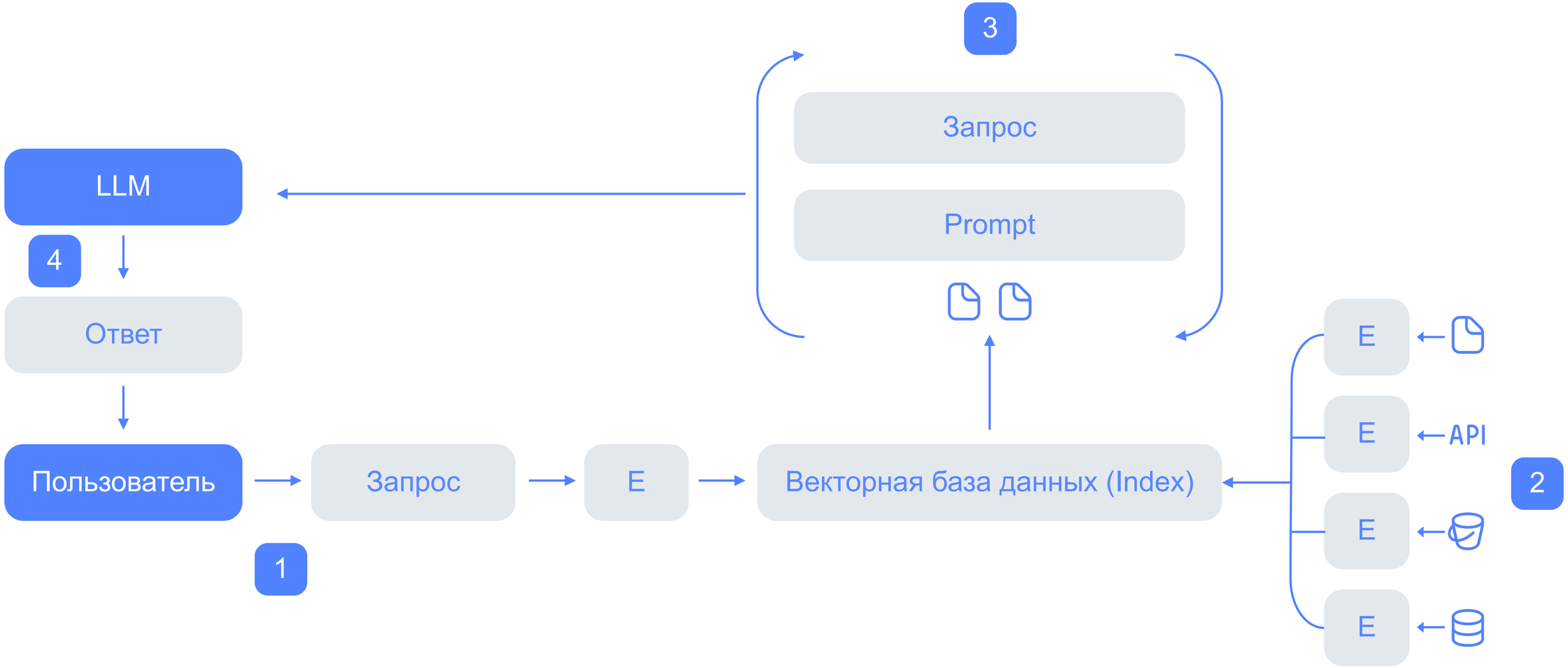
Чтобы дообучить модель YandexGPT нужно подготовить файл с формате JSON. Файл надо сохранить в кодировке UTF-8:

```
[
  {
    "request": [
      {
        "role": "system",
        "text": "Текст инструкции"
      },
      {
        "role": "user",
        "text": "Текст запроса"
      }
    ],
    "response": "Ожидаемый ответ"
  }
]
```

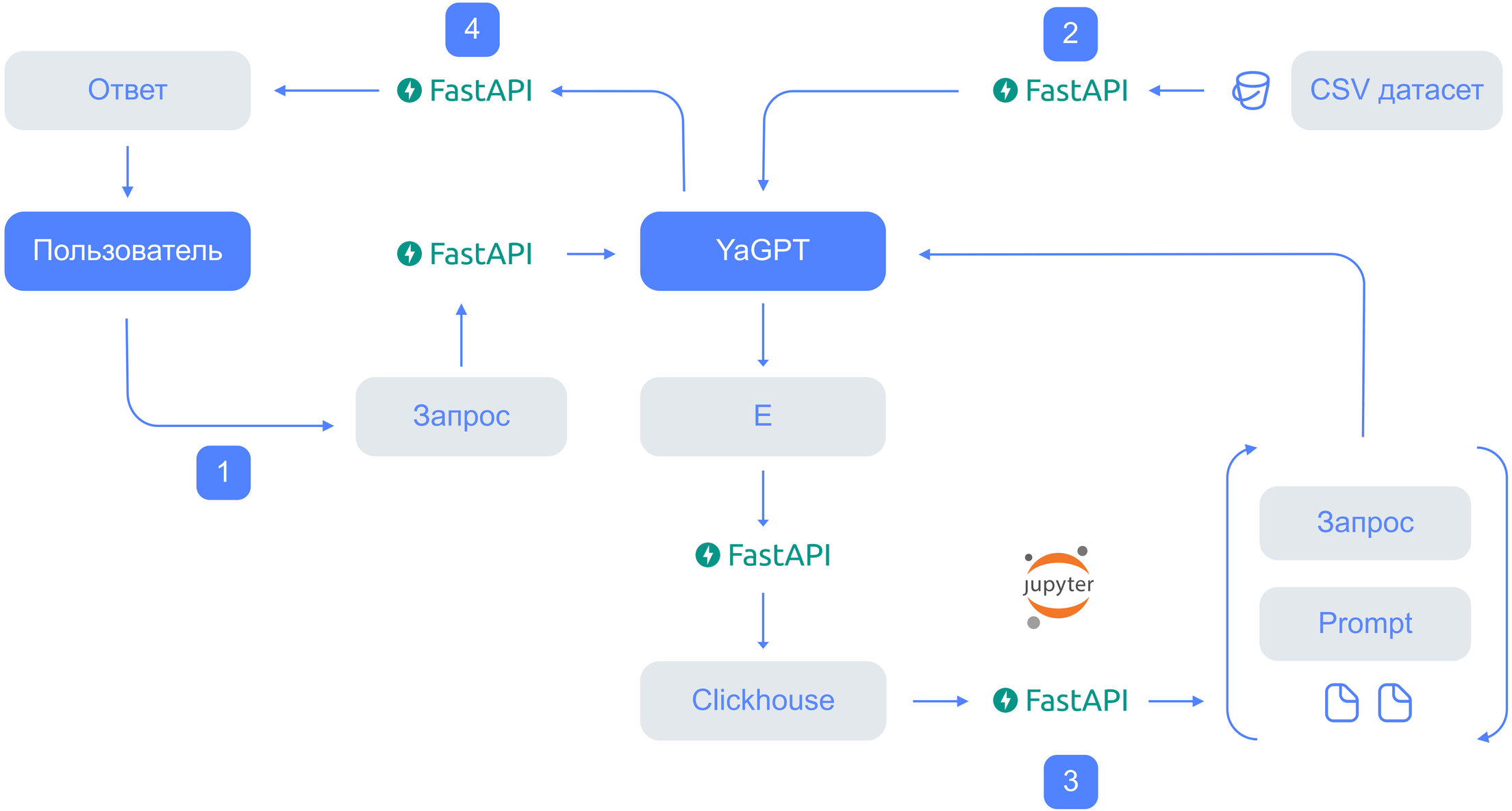
Файл может содержать не больше 10 000 вопросов и ответов. Максимальная длина запроса — 4000 символов, максимальная длина эталонного ответа — 2000 символов



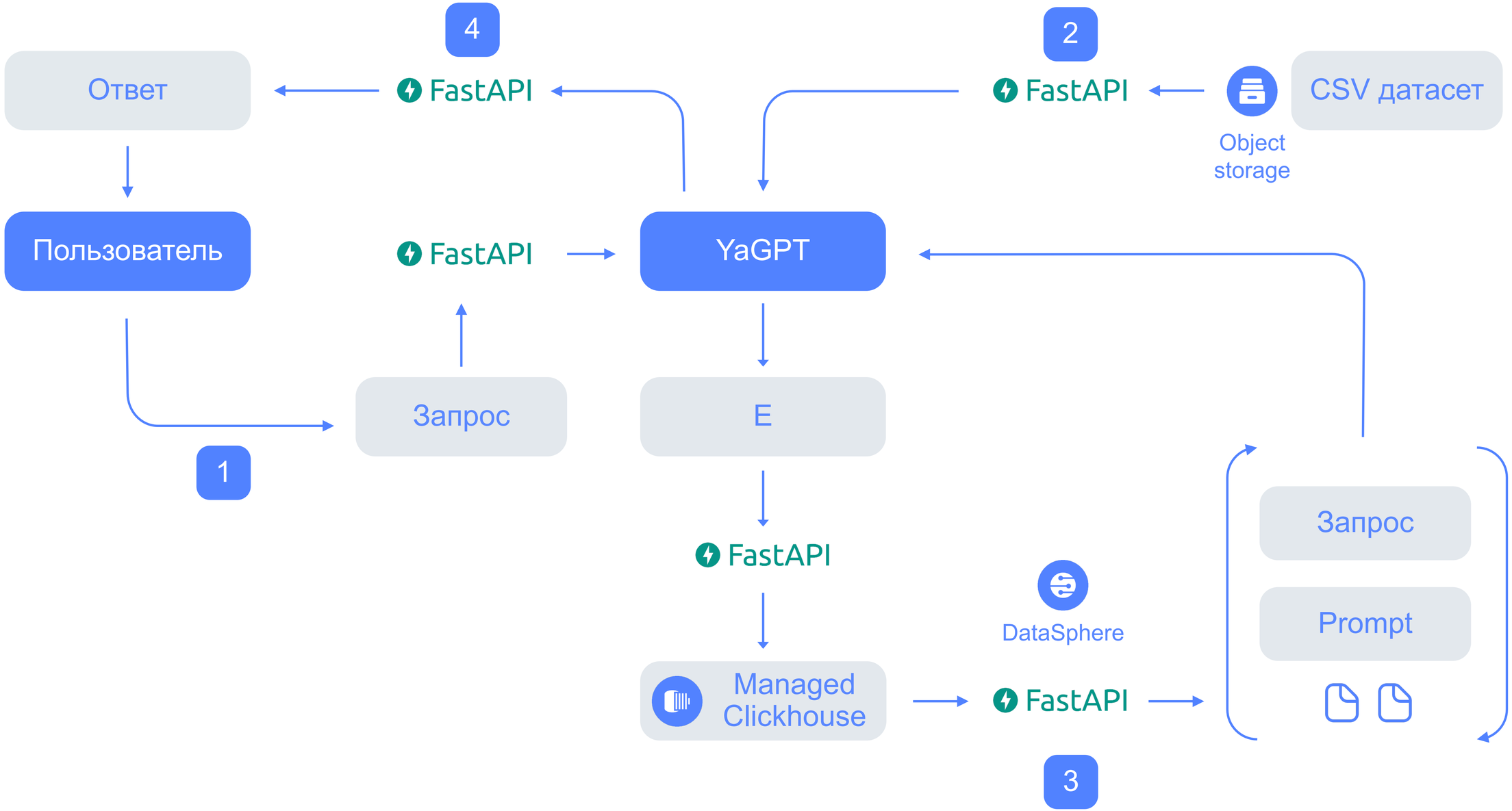
Архитектура RAG



Архитектура RAG on-premise



Архитектура RAG YC



ClickHouse — когда использовать в качестве БД для хранения векторов?

Одновременный поиск
по векторам и фильтрации
по метаданным
и/или агрегации, JOIN

Линейное сопоставление
по очень большим объемам
векторов

Нужна поддержка SQL

Нет необходимости выполнять
векторизацию на стороне БД

Уже есть ClickHouse
в используемом стеке

ClickHouse — когда не использовать в качестве векторной БД?

Weaviate

1. Если ваш датасет небольшой и полностью помещается в памяти
2. Нет метаданных и нужно только сопоставление по векторам
3. Нужна оцифровка на стороне БД в момент вставки

ClickHouse — основные функции

cosineDistance (vector1, vector2)

L2Distance (vector1, vector2)

Сценарий 2 – операция “Обновить Pytorch”

Команда разработчиков

PyTorch

Версия библиотеки 2.3.0

Существующие фреймворки

Для продукт-менеджеров

Langflow (langflow.org)

Flowise (flowiseai.com)

Rivet (rivet.ironcladapp.com)

---

n8n (n8n.io)

---

create-llama (npmjs.com/package/create-llama)

Существующие фреймворки

Для программистов

LLM chains	Multi-Agent frameworks
Langchain	AutoGEN
LlamaIndex	AutoGPT
Haystack	GPT-engineer
Semantic Kernel from	BabyAGI
Griptape	Langroid
	CrewAI

Полезные ссылки

github.com/pytorch/pytorch/releases/tag/v2.3.0

---

python.langchain.com/v0.1/docs/integrations/vectorstores/clickhouse

---

python.langchain.com/v0.1/docs/integrations/text_embedding/yandex

---

clickhouse.com/blog/vector-search-clickhouse-p1

---

python.langchain.com/v0.1/docs/integrations/llms/yandex

Практика!

Кузьма Лешаков
Архитектор Data Platform,
Yandex Cloud

github.com/techkuz/fast-api-rag-clickhouse-yc