Подготовка данных для RAG

Автор: Ганеев Рустам Марсович

План лекции

1. Основы

GIGO и Ландшафт данных

2. Web Scraping

Браузеры, API, Anti-bot

3. Parsing

PDF, OCR, Docling, Tables

4. Cleaning

PII, Нормализация, Дедупликация

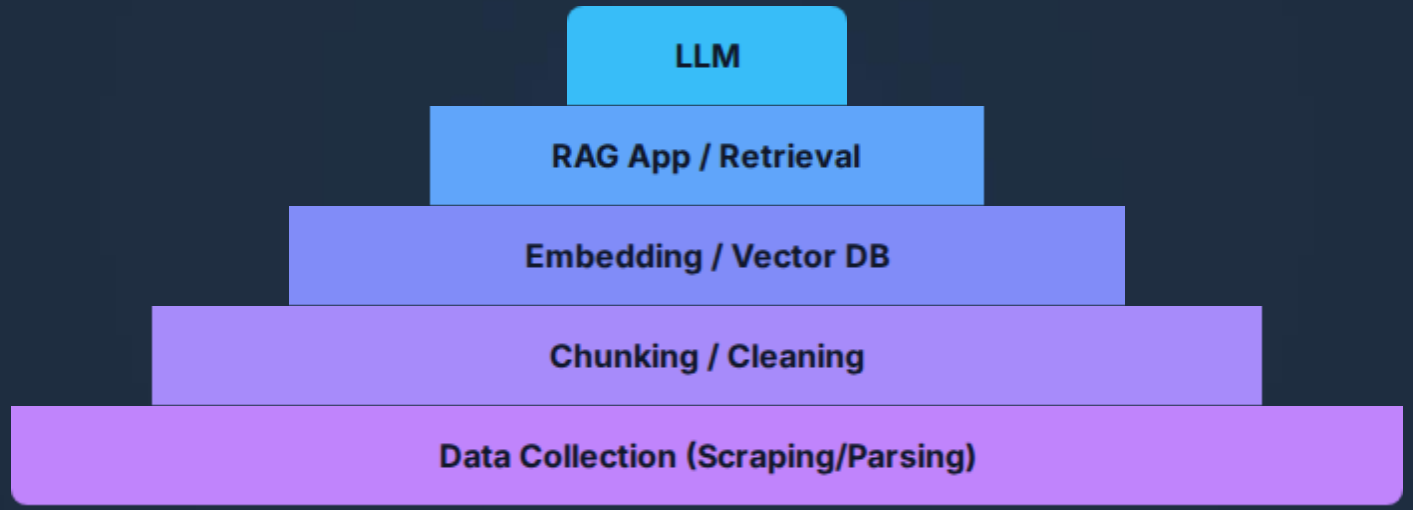
Принцип GIGO

Garbage In — Garbage Out

Ваш RAG умный ровно настолько, насколько чисты ваши данные.

- HTML теги в промпте сбивают модель.
- Ошибки OCR приводят к потере имен собственных.
- Некорректные таблицы создают ложные факты.

Пирамида потребностей RAG



90% ошибок случаются на нижнем уровне.

Ландшафт Данных

Структурированные

SQL, CSV, JSON,

API.

Легко читать, сложно получить контекст.

Неструктурированные

PDF, HTML, Video, Audio.

Богатый контекст, сложно извлечь.

Основы HTTP: Request/Response

Анатомия запроса

- **URL:** Адрес ресурса.
- **Method:** GET (дай), POST (возьми).
- **Headers:** "Я iPhone", "Я Chrome".
- **Cookies:** "Я залогинен".

```
GET /article/123 HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0...
Accept: text/html

HTTP/1.1 200 OK
Content-Type: text/html...
```

Селекторы: XPath vs CSS

Как найти нужный элемент?

Тип	Пример	Плюсы/Минусы
CSS	[class*=”btn-block”]	Быстро, просто. Не умеет "назад".
XPath	//div[@class='content']/p[contains(text(),'Price')]	Можно искать по тексту. Медленнее.

Статика vs Динамика (SPA)

Static Site

Сервер отдает готовый HTML.

Инструмент: requests, BeautifulSoup

Скорость: 100 мс

Dynamic (SPA)

HTML пустой. Данные грузит JS (React/Vue).

Инструмент: Playwright, Selenium

Скорость: 2-5 сек

Скрытые API

Динамические сайты откуда-то берут данные. Обычно это JSON API.

Метод: Открыть DevTools -> Network -> XHR/Fetch.

Если найти этот эндпоинт, можно парсить данные в чистом JSON, минуя HTML и браузер.

Уровни защиты от ботов

Уровень 1

Проверка User-Agent. Блок по IP.

Уровень 2

Проверка JS (есть ли `window.navigator.webdriver`).

Уровень 3

CAPTCHA (ReCaptcha, hCaptcha).

Уровень 4

TLS Fingerprinting (JA3) и поведенческий анализ.

Почему PDF — это боль?

PDF создан для **принтера**, а не для экрана.

Внутри PDF нет слов "абзац" или "таблица". Есть инструкции: "Нарисуй букву 'A' по координатам X=10, Y=20".

Проблемы:

- Чтение многоколоночного текста.
- Склеивание слов (kerning).
- Невидимый текст (Watermarks).

Извлечение таблиц

Таблицы — главный источник данных в бизнесе. Старые парсеры превращают их в кашу.

Lattice (Линии)

Ищет черные линии-границы ячеек. Работает только для таблиц с сеткой.

Stream (Пробелы)

Ищет белые промежутки между колонками. Ненадежно.

Vision (AI)

Компьютерное зрение видит структуру как человек. (Используется в Docling/LlamaParse).

IBM Docling: Новый Стандарт

Docling — это библиотека от IBM Research.

Она объединяет Rule-based подходы и AI Vision модели.

Философия: Любой документ на входе -> Markdown на выходе.

Это open-source и работает локально.

Альтернатива: LlamaParse

LlamaParse (SaaS)

Сервис от создателей LlamaIndex.

Плюсы: Идеально парсит сложнейшие таблицы, имеет режим "GPT-4o mode" для описания картинок.

Минусы: Платно, данные уходят в облако.

Альтернатива: Unstructured.io

Unstructured (Modular)

Огромный комбайн с коннекторами ко всему (S3, Google Drive, Slack).

Разбивает документ на элементы: Title, NarrativeText, Table.

Позволяет гибко настраивать стратегии (Hi-Res, Fast).

Сравнение Парсеров

Инструмент	Тип	Качество таблиц	Цена
Docling	Local / Open Source	⭐ ⭐ ⭐ ⭐	Бесплатно (CPU/GPU)
LlamaParse	SaaS API	⭐ ⭐ ⭐ ⭐ ⭐	$$$ (постранично)
Unstructured	Hybrid	⭐ ⭐ ⭐	Free / $$$
PyPDF2	Local Lib	⭐	Бесплатно

Технологии OCR

Что делать, если PDF — это скан?

Tesseract

Старый стандарт. Требует препроцессинга (контраст, поворот).

EasyOCR / Paddle

Deep Learning модели. Лучше работают с текстом на фоне.

VLM (GPT-4o)

Отправляем картинку страницы в LLM. Дорого, но идеально.

Офисные форматы (DOCX, PPTX)

Эти форматы — на самом деле ZIP-архивы с XML внутри.

Совет: Не используйте конвертацию в PDF перед парсингом! Вы теряете структуру.

Парсите XML напрямую (через `python-docx` или Docling) — так вы сохраните иерархию заголовков и списков идеально.

Архитектура ASR (Whisper)

OpenAI Whisper

Модель типа Encoder-Decoder Transformer.

Обучена на 680,000 часах аудио.

Фичи: Мультиязычность, таймстемпы, определение языка.

Можно запускать облегченные версии (`distil-whisper`) для скорости.

Диаризация (Кто говорит?)

Процесс разделения аудио на дорожки спикеров.

[Speaker A]: Добрый день, у меня проблема.

[Speaker B]: Здравствуйте, опишите детали.

Инструменты:

- **Pyannote.audio:** Лучшая Open Source библиотека (HuggingFace).
- **NVIDIA NeMo:** Для энтерпрайз решений.

VAD (Voice Activity Detection)

Прежде чем отправлять аудио в Whisper, нужно удалить тишину.

В звонках 30-50% времени — это молчание.

Silero VAD: Сверхбыстрая нейросеть (запускается на CPU), которая находит границы речи. Экономит ресурсы видеокарты в 2 раза.

Индексация Видео

Audio Track

ASR + Диаризация.

OCR on Frames

Распознавание текста на слайдах в видео.

Visual Captioning

Использование VLM для описания происходящего на экране (раз в 5 сек).

Нормализация Текста

Приведение текста к каноническому виду.

- **Unicode Normalization (NFKC):** Превращает разные виды кавычек и пробелов в стандартные.
- **Whitespace:** Замена \n\n\n на \n.
- **Mojibake:** Исправление битых кодировок (ftfy library).

Вопросы?

Спасибо за внимание!