Ограничения stand-alone LLM моделей

Светлана Каримова

Контекст курса

Пройденные темы:

- Лекция 1: Что такое LLM
- Лекция 2: PEFT и LoRA
- Лекция 3: Оценка качества и бенчмарки

Сегодня: Границы возможностей

Следующая лекция: Многофункциональный сервис на базе LLM

Цели лекции

- Понять границы применимости базовых LLM
- Изучить способы расширения возможностей
- Познакомиться с концепцией агентности
- Научиться определять, когда нужны дополнительные функции

Что такое stand-alone LLM?

Определение:

модель "как есть" без внешних инструментов

Базовый паттерн:

prompt → LLM → response

Примеры:

- ChatGPT без плагинов
- Claude без tools
- Локальная Llama

Основные ограничения

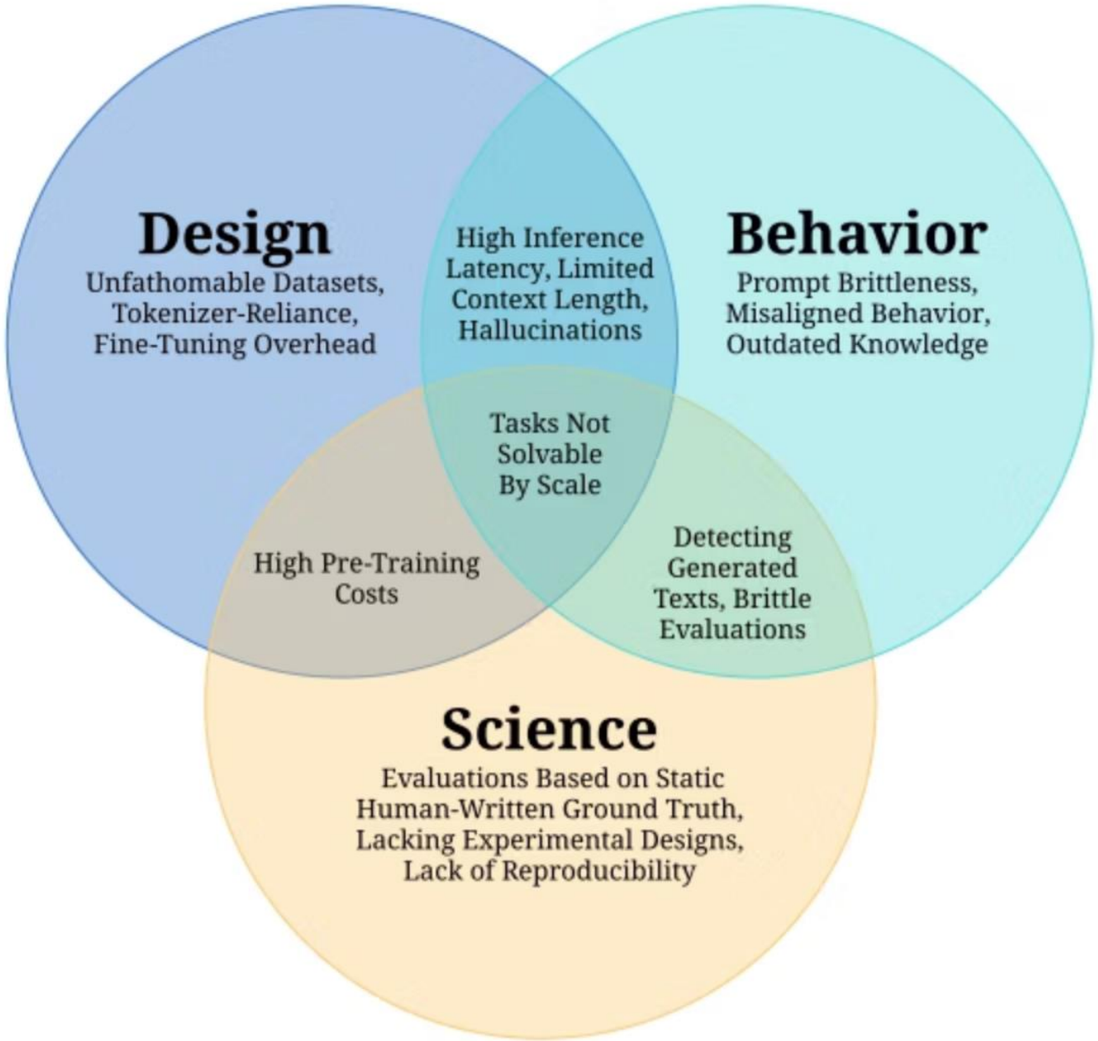
Карта ограничений

Категории ограничений:

1. Временные (знания)
2. Вычислительные
3. Архитектурные
4. Модальностные
5. Интерактивные

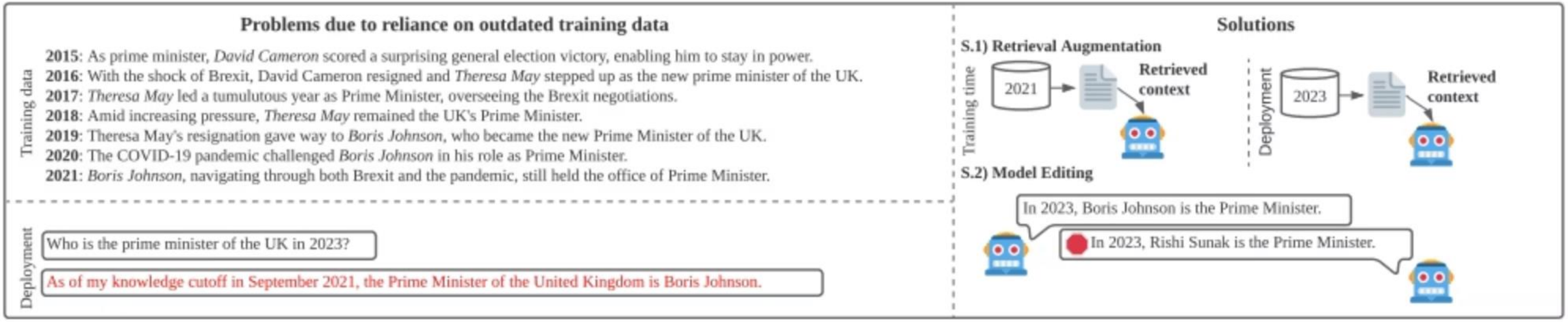
Survey: Challenges and Applications of Large Language Models

Kaddour et al., 2023 | https://arxiv.org/abs/2307.10169



Ограничение 1: Устаревшие знания

- Модель обучена на данных до определенной даты (cutoff date)
- Пример: GPT-4 (2023) не знает события 2024 года
- Не может обновить знания без переобучения
- Критично для: новостей, актуальных цен, свежих исследований



Ограничение 2: Нет доступа к приватным данным

Модель обучена на публичных данных

Не знает:

- Документы вашей компании
- Базу данных клиентов
- Внутренние процедуры

Решение через fine-tuning имеет свои проблемы

Date	Name	Size		Sources	Public
		GB	Tokens*		
2014	BookCorpus [684, 36]	5 GB	11 B	Novels	Yes
2019	OSCAR [399]	6.3 T	?	Webpages in 166 languages	Yes
2019	WebText [440]	40 GB	?	Webpages	No
12.2020	CC-100 [100]	2.5 TB	292 B	Webpages in 100 Languages	Yes
12.2020	The Pile [165, 41]	825 GB	300 B	Science, Webpages, GitHub Code, Law, etc.	Yes
2020	C4 [443]	745 GB	156 B	Webpages	Yes
10.2020	mC4 [631]	?	6.3 T	Webpages in 101 Languages	Yes
2021	MassiveText [441]	10.5 TB	2.34 T	Webpages, Books, News, and Code	No
12.2021	GLaM [130]	?	1.6 T	Webpages, Wikipedia, Conversations, Forums, Books, News	No
01.2022	Infiniset [551]	?	2.81 T	Forum dialogs, C4 data, Code, Wikipedia, Webpages	No
06.2022	ROOTS [289]	1.61 TB	2.34 T	Webpages in 46 languages and GitHub Code in 13 languages	Yes
11.2022	The Stack [271]	6 TB	235 B	GitHub Code in 30 languages	Yes
04.2023	LLaMA [556] / RedPajama [98]	2.7 TB	1.2 T	Webpages, GitHub Code, Science, Wikipedia, Books	Yes
06.2023	RefinedWeb [415]	2.8 TB	600 B	Webpages	Yes

Ограничение 3: Нет детерминированных вычислений

LLM плохо справляется с:

- Точными математическими вычислениями
- Манипуляциями с большими числами
- Криптографией

Пример: "Сколько будет 8,547 × 37,129?" → неточный ответ (особенно при использовании по API)

Причина: Модель предсказывает токены, а не вычисляет

Исследование: Faith and Fate: Limits of Transformers on Compositionality

Dziri et al., 2023 | https://arxiv.org/abs/2305.18654

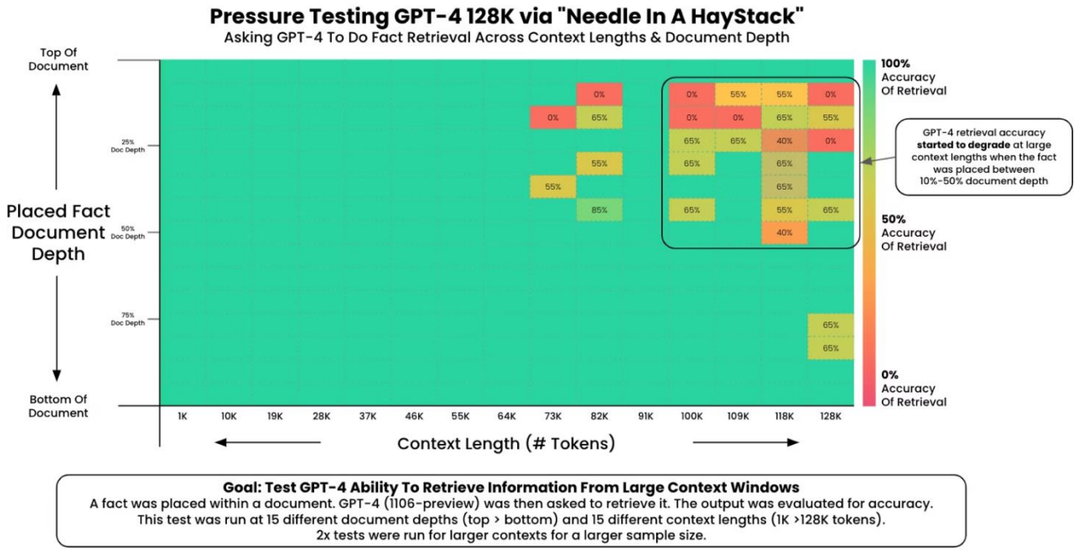
Ограничение 4: Длина контекста

Ограничение context window (например, 4K, 8K, 128K токенов)

Не может обработать:

- Очень длинные документы целиком
- Большие кодовые базы
- Историю длинной переписки

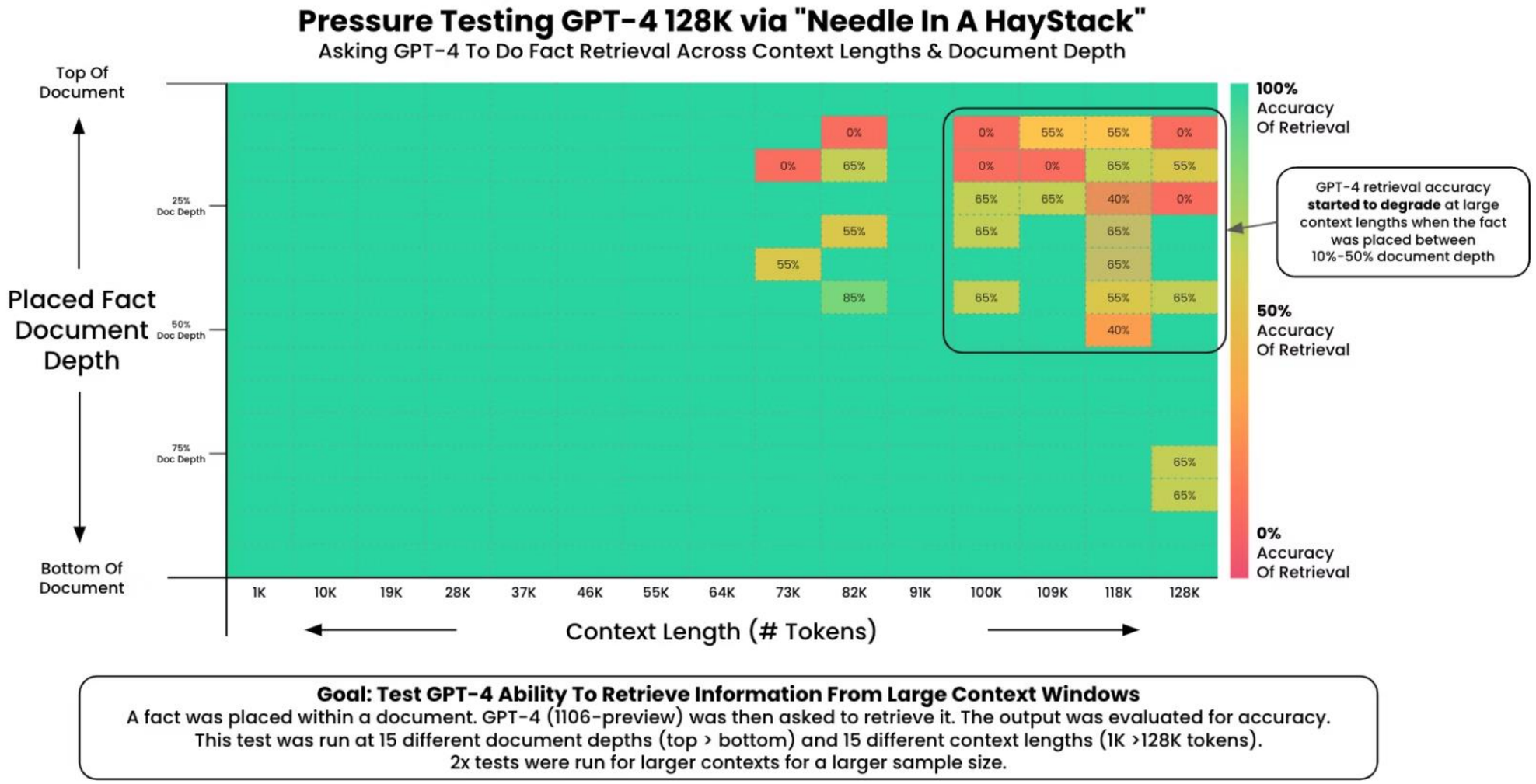
Стоимость растет квадратично с длиной контекста (attention)



Бенчмарк: Needle In A Haystack (Greg Kamradt, 2023)

https://github.com/gkamradt/LLMTest_NeedleInAHaystack

Ограничение 4: Длина контекста



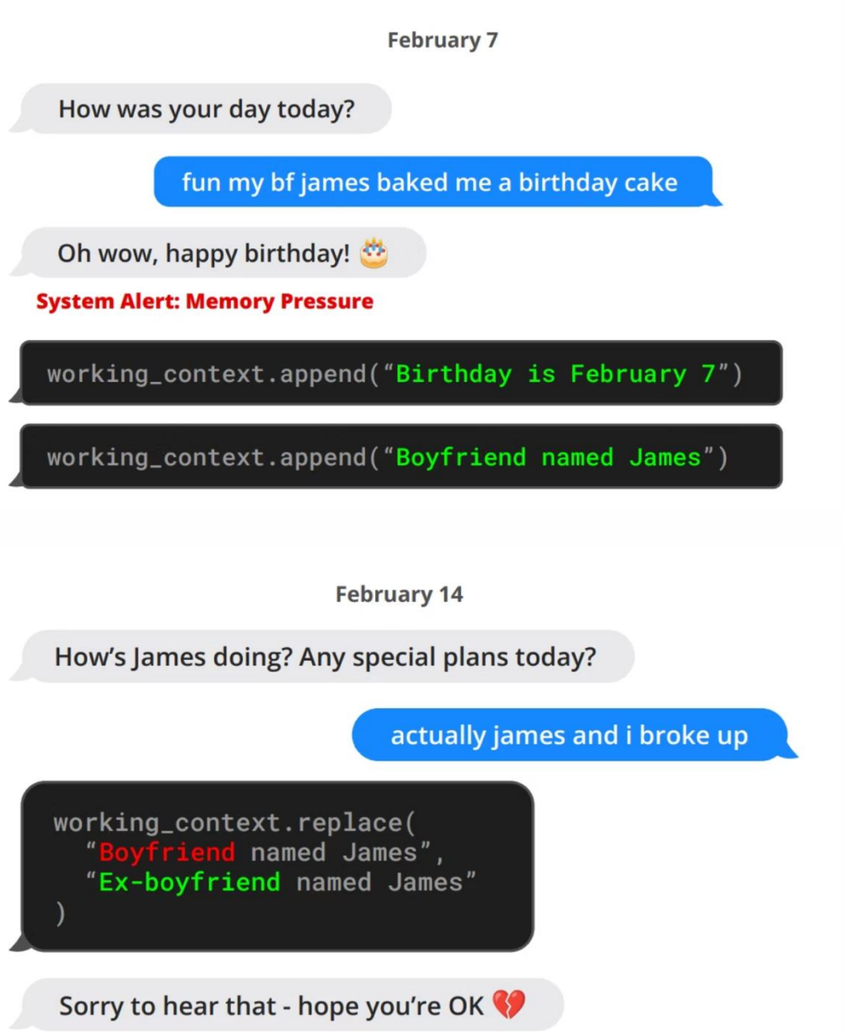
Бенчмарк: Needle In A Haystack (Greg Kamradt, 2023) https://github.com/gkamradt/LLMTest_NeedleInAHaystack

Ограничение 5: Нет долговременной памяти

- Каждый запрос — изолированная транзакция
- Нет сохранения состояния между сессиями
- Для поддержания контекста нужно передавать всю историю
- Растут затраты на токены

Исследование: MemGPT: Towards LLMs as Operating Systems

Packer et al., 2023 | https://arxiv.org/abs/2310.08560



Ограничение 6: Мономодальность

Базовые LLM — только текст

Не могут:

- Анализировать изображения (в text-only моделях)
- Воспроизводить аудио
- Генерировать видео

Мультимодальные модели частично решают проблему, но не все

Ограничение 7: Нет взаимодействия с внешним миром

Не может:

- Открыть браузер и загрузить страницу
- Записать данные в базу
- Сделать API запрос
- Отправить email

Это только текстовая модель, а не система действий

Ограничение 8: Галлюцинации

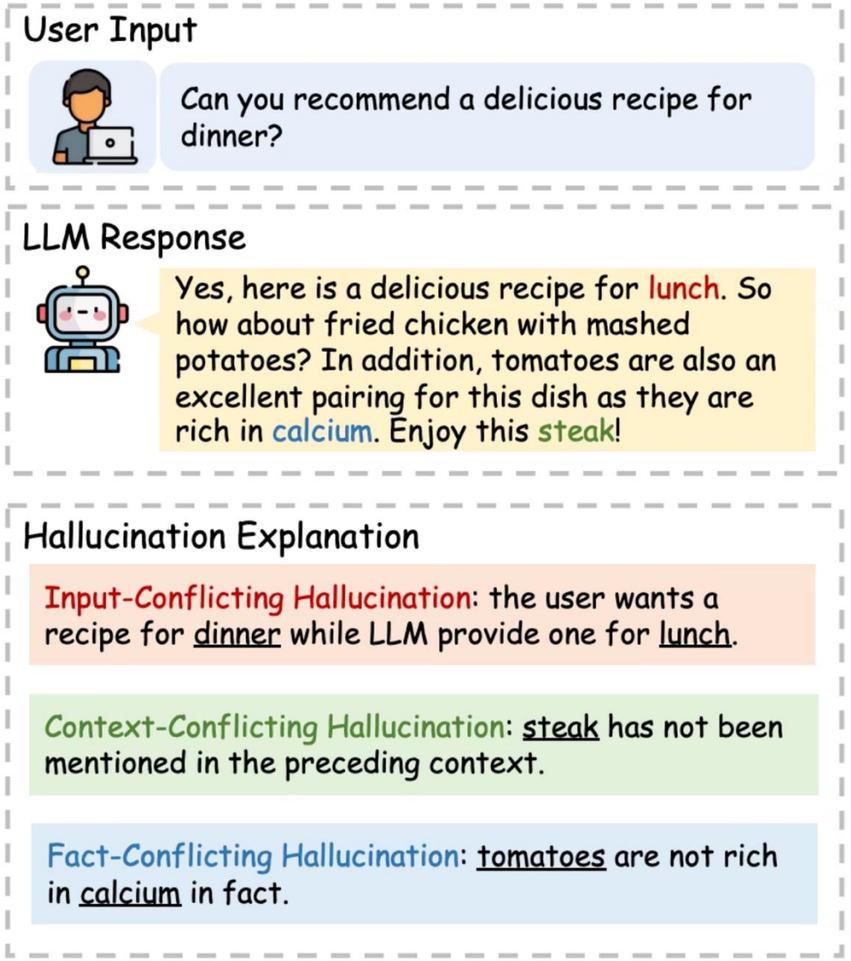
Модель может генерировать ложную информацию с уверенным видом

Особенно при:

- Отсутствии знаний
- Неоднозначных вопросах
- Запросах на фактическую точность

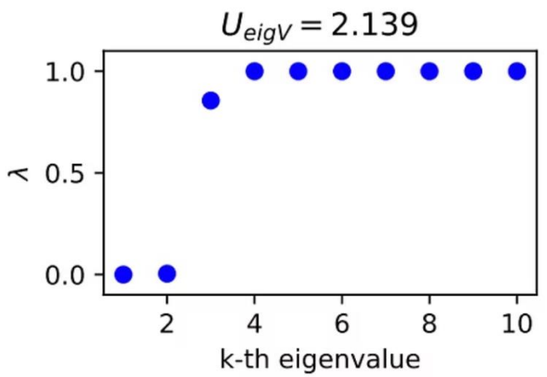
Survey: Siren's Song in the AI Ocean: A Survey on Hallucination in LLMs

Zhang et al., 2023 | https://arxiv.org/abs/2309.01219

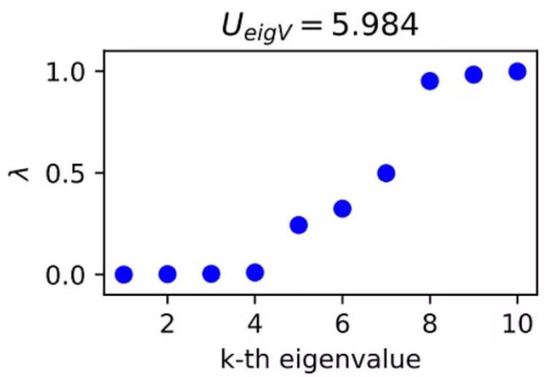


Ограничение 9: Недетерминированность

- Одинаковый промпт → разные ответы
- Temperature и sampling создают вариативность
- Проблема для задач, требующих стабильности
- Даже при temperature=0 возможны различия




```
Pink Floyd
Pink Floyd in Edinburgh
Pink Floyd
Shambles
Pink Floyd
Pink Floyd
Pink Floyd
Pink Floyd
Pink Floyd
Pink Floyd
Pink Floyd
```




```
Leukemia
Low Blood Pressure
Cervical cancer
Cancer in 1953 at 41
Breast cancer
Tuberculosis
Cancer
Leukaemia
Cancer (in 1953 at age 41)
Throat cancer
```

Одна из статей использует теорию графов, чтобы понять, насколько «путается» нейросеть - чем выше *lambda*, тем менее разнообразные ответы дает модель

Статья: On the Reliability of LLMs: Uncertainty Quantification

Xiong et al., 2023 | https://arxiv.org/abs/2305.19187

Ограничение 10: Стоимость и латентность

- Каждый запрос стоит денег (за токены)
- Задержка ответа (latency) — секунды
- Масштабирование дорого
- Не подходит для real-time задач с низкой латентностью

Статья: Efficiently Scaling Transformer Inference

Pope et al., 2022 | https://arxiv.org/abs/2211.05102

Итоги: карта ограничений

Временные:

Устаревшие знания, нет приватных данных

Вычислительные:

Плохие вычисления, длина контекста, стоимость

Архитектурные:

Нет памяти, недетерминированность, галлюцинации

Модальностные:

Мономодальность

Интерактивные:

Нет действий во внешнем мире

Способы преодоления ограничений

Стратегии преодоления

Три основных подхода:

01

Расширение через внешние инструменты (Tools/Functions)

02

Расширение через внешние данные (RAG, Memory)

03

Мультимодальность

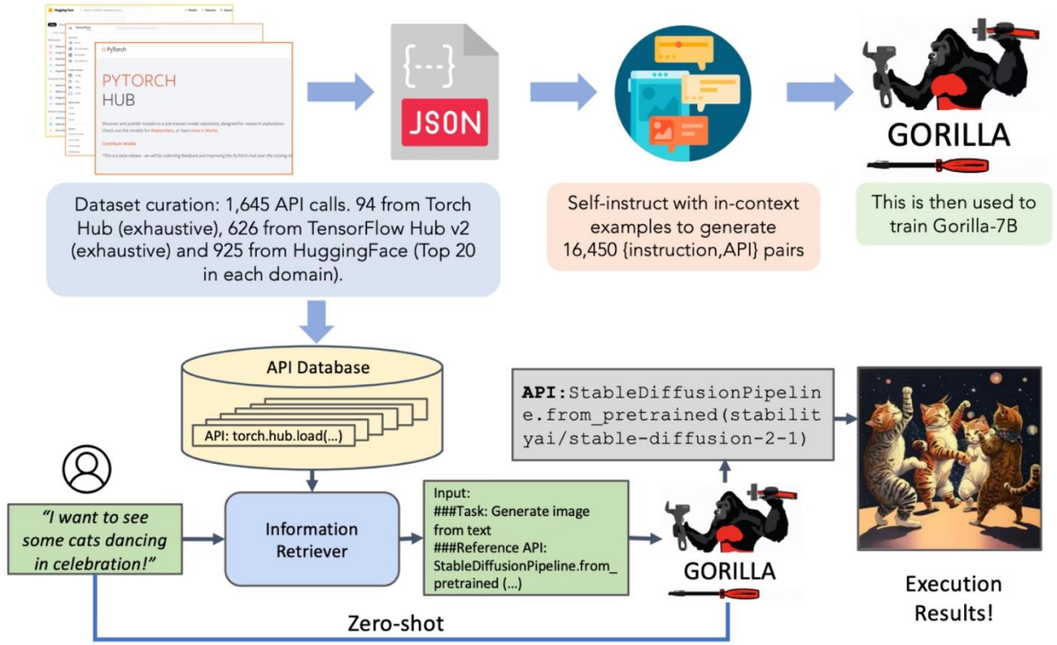
Комбинирование подходов

Подход 1: Tool Use (Function Calling)

Концепция: LLM решает, КОГДА и КАКОЙ инструмент вызвать

Процесс:

1. Описываем доступные функции в промпте
2. LLM анализирует запрос и выбирает функцию
3. Система выполняет функцию
4. Результат возвращается в LLM для формулировки ответа



Статьи:

- Toolformer (Meta AI, 2023) | https://arxiv.org/abs/2302.04761
- Gorilla: Large Language Model Connected with Massive APIs Patil et al., 2023 | https://arxiv.org/abs/2305.15334

Примеры инструментов

Вычисления:

- Калькулятор
- Python-интерпретатор

Поиск:

- Веб-поиск
- Поиск по документам

Данные:

- SQL-запросы
- API внешних сервисов

Действия:

- Отправка email
- Создание задачи

Актуальность:

- Получение текущей даты/времени

Пример: преодоление ограничения вычислений

1	User: "Сколько будет 8547 × 37129?"
2	LLM: *определяет, что нужен калькулятор*
3	Function call: calculate(8547 * 37129)
4	System: *выполняет* → 317,379,663
5	LLM: "Результат умножения 8,547 на 37,129 равен 317,379,663"

Пример: преодоление ограничения актуальности

1	User: "Какая погода сейчас в Москве?"
2	LLM: *определяет, что нужен API погоды*
3	Function call: get_weather(city="Москва")
4	System: *вызывает API* → {temp: -5, condition: "снег"}
5	LLM: "В Москве сейчас -5°C, идет снег"

Реализация Tool Use

Платформы:

- OpenAI Functions/Tools API
- Anthropic Claude Tools
- Open-source: LangChain, LlamaIndex

Схема:

- Описание функции в формате JSON Schema
- LLM возвращает structured output с именем функции и параметрами
- Ваш код выполняет функцию

Подход 2: Retrieval-Augmented Generation (RAG)

Проблема: модель не знает приватные данные

Решение: добавить релевантный контекст в промпт

Процесс:

1. Индексация документов (векторные embeddings)
2. Поиск релевантных фрагментов по запросу
3. Добавление фрагментов в промпт
4. LLM отвечает на основе этого контекста

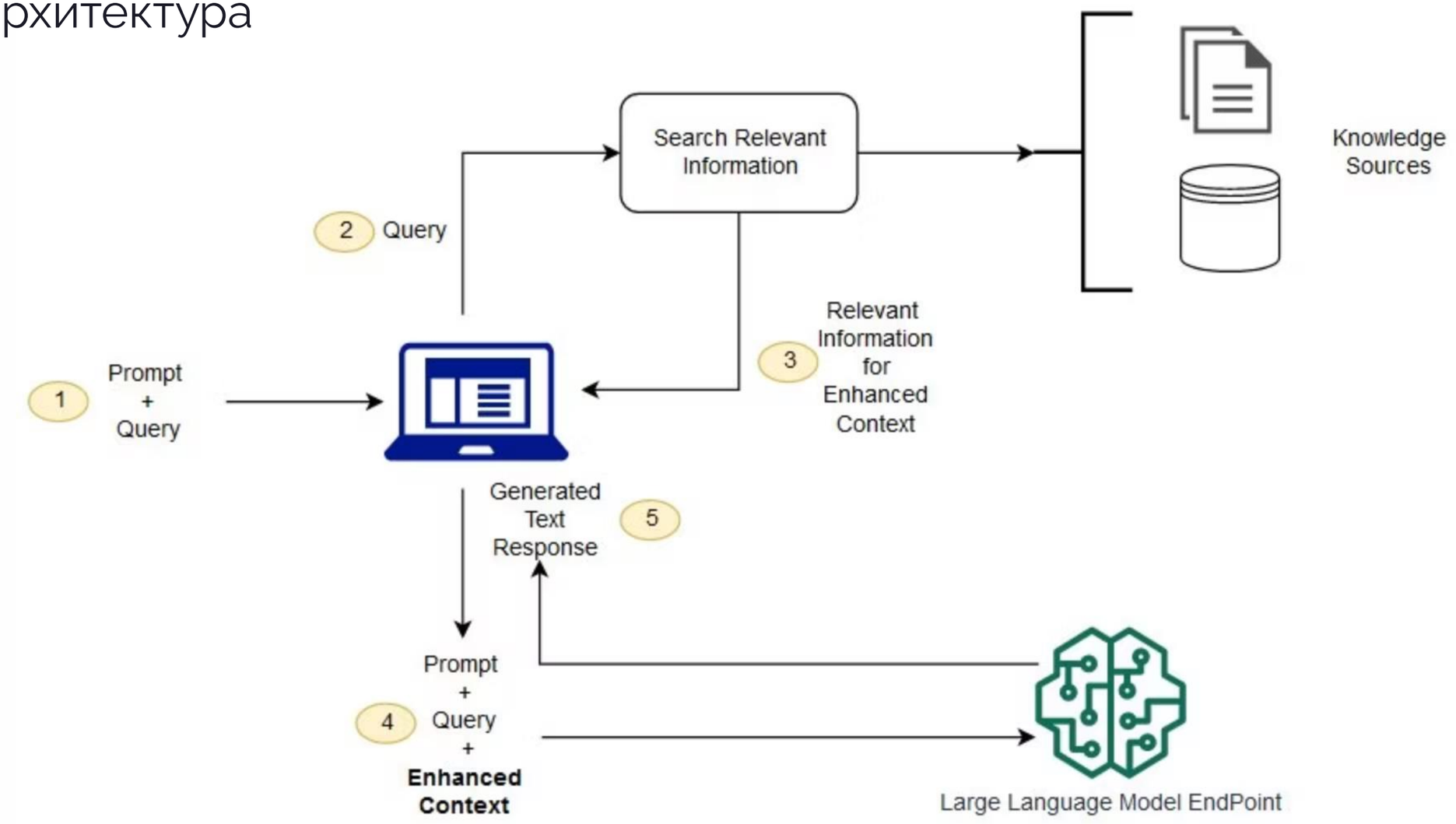
Статья: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Lewis et al., 2020 | https://arxiv.org/abs/2005.11401

Fun fact (аналогия):

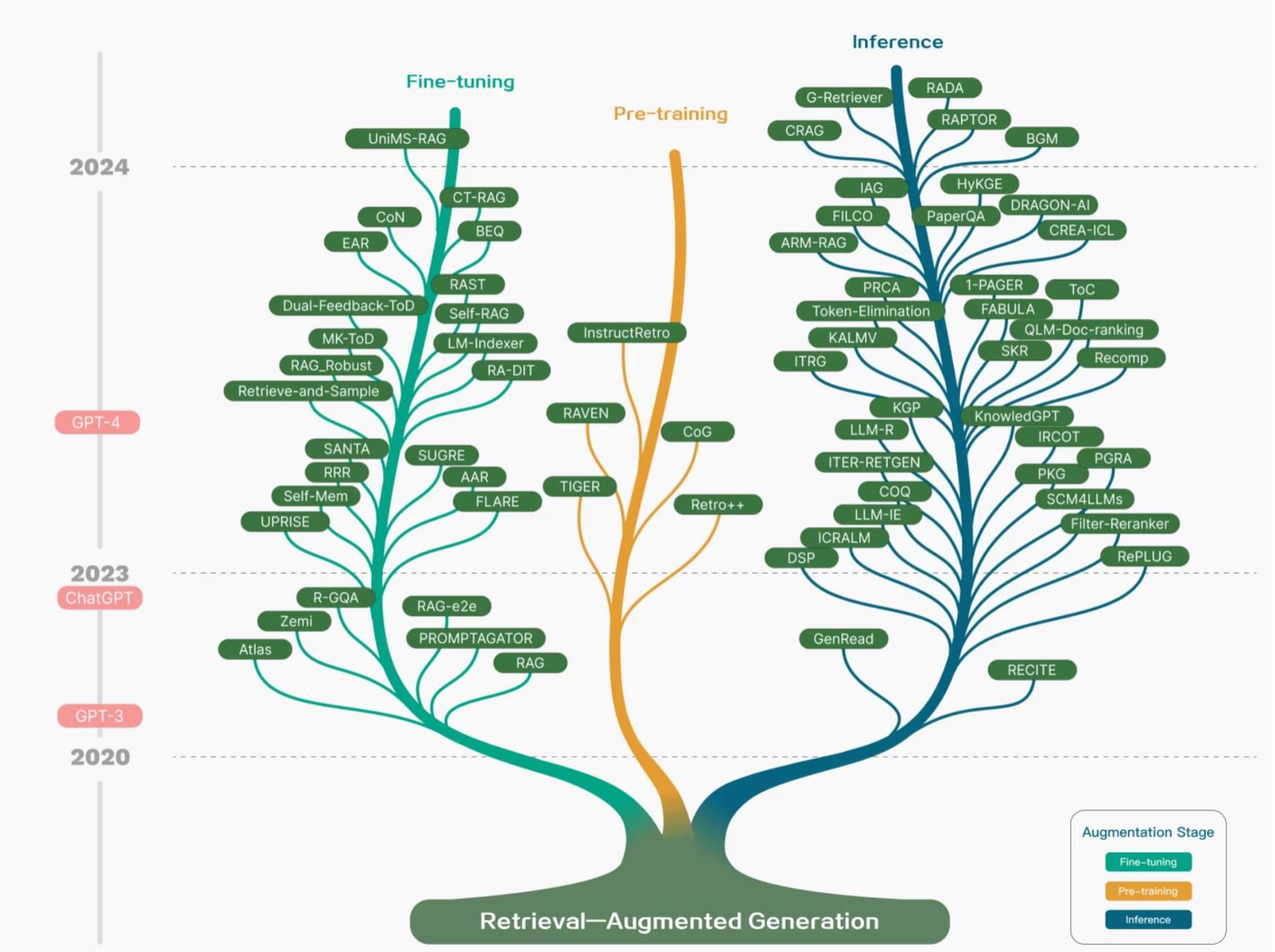
- В 2011 году суперкомпьютер Watson в прямом эфире сразился с двумя величайшими чемпионами шоу — Кеном Дженнингсом и Брэдом Раттером. Watson не просто победил, он **разгромил** их, заработав $77,147 против $24,000 у ближайшего соперника.
- Во время игры Watson **не был подключен к сети**. Все 200 миллионов страниц текста (включая всю Википедию) были закачаны в его оперативную память (15 ТБ RAM).

RAG: архитектура



Преимущества RAG

- Нет необходимости в fine-tuning
- Легко обновлять базу знаний
- Снижение галлюцинаций (есть source of truth)
- Масштабируется на большие объемы документов
- Можно указывать источники в ответе



Survey: Retrieval-Augmented Generation for Large Language Models: A Survey

Gao et al., 2023 | https://arxiv.org/abs/2312.10997

Ограничения RAG

- Качество поиска критично
- Нужно оптимизировать chunking (разбиение текста)
- Проблема с очень большим количеством документов
- Может не найти информацию, если она распределена по документам

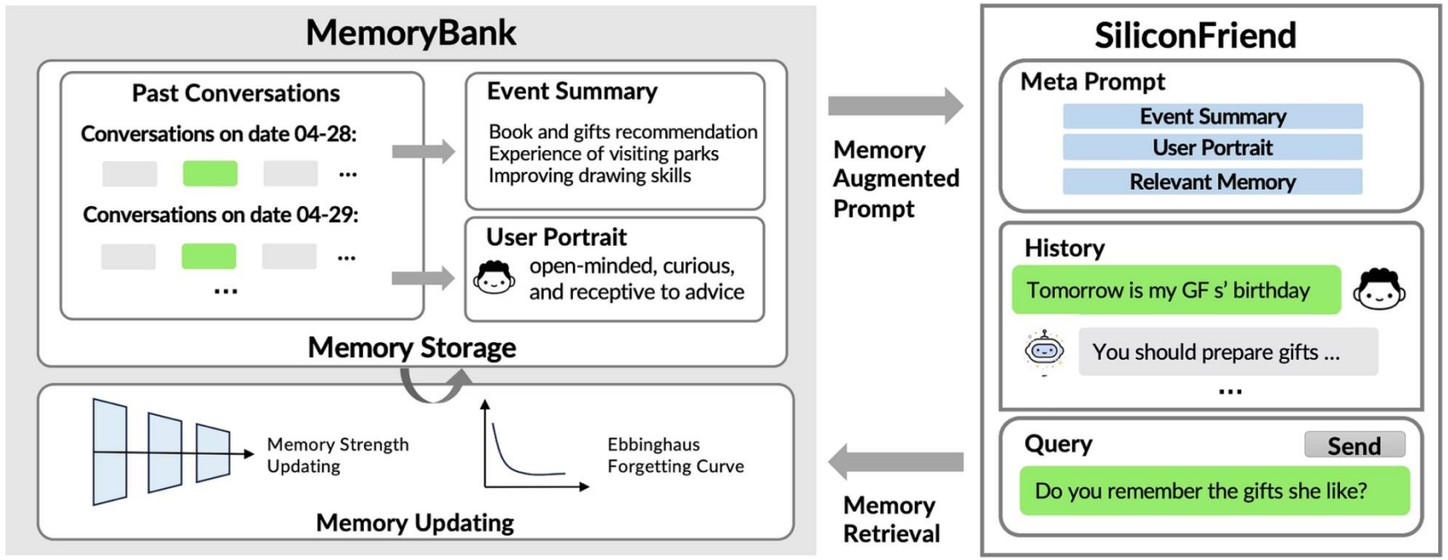
Подход 3: Memory Systems

Проблема: нет долговременной памяти

Решение: внешнее хранилище памяти

Типы:

- **Short-term:** хранение текущей сессии
- **Long-term:** хранение между сессиями →
- **Entity memory:** факты о сущностях (люди, места)
- **Semantic memory:** обобщенные знания



Статья: MemoryBank: Enhancing LLMs with Long-Term Memory
Zhong et al., 2023 | https://arxiv.org/abs/2305.10250

Мультимодальность

Расширение модальностей:

Vision:

- Анализ изображений (GPT-4V, Claude Sonnet 3.5)

Audio:

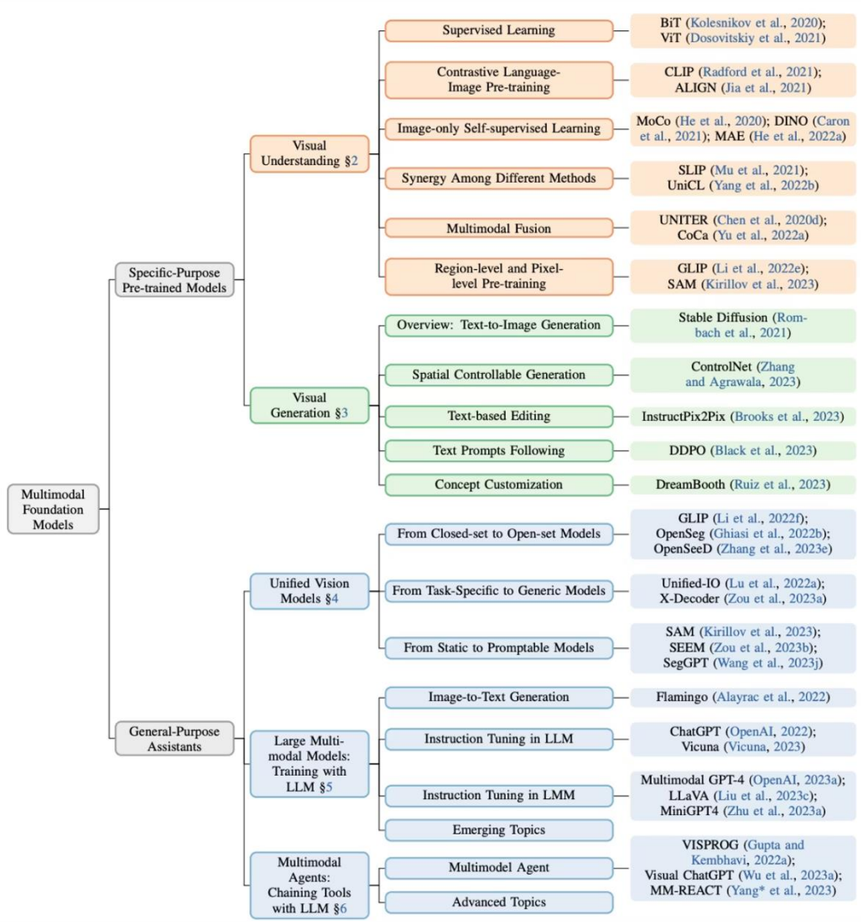
- Распознавание и генерация речи

Video:

- Понимание видео (developing)

Позволяет работать с богатым контекстом

Критично для многих применений (медицина, робототехника)



Survey: Multimodal Foundation Models: From Specialists to General-Purpose Assistants

Li et al., 2023 | https://arxiv.org/abs/2309.10020

Комбинирование подходов

Реальные системы используют все подходы вместе:

RAG

для доступа к документам

Tools

для действий и вычислений

Memory

для персонализации

Multimodal

для богатого ввода

Пример: Ассистент компании с доступом к базе знаний, CRM и email

Следующий уровень: от функций к системам

Эволюция сложности

Уровень 1: Stand-alone LLM

Простой запрос-ответ

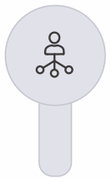
Уровень 2: LLM + Tools

Однократное использование инструментов

Уровень 3: LLM + RAG + Memory

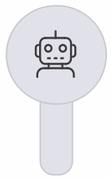
Контекстуализированные ответы с памятью

На этой лекции мы тут



Уровень 4: Многофункциональные сервисы

Комбинация всех подходов, оркестрация



Уровень 5: Агентные системы

Автономное многошаговое планирование

Когда какой подход использовать?

Stand-alone LLM:

Простые текстовые задачи без актуальных данных

Tools:

Нужны вычисления или действия (API, расчеты)

RAG:

Нужен поиск по документам/базе знаний

Memory:

Важен контекст предыдущих взаимодействий

Multimodal:

Работа с изображениями, аудио, видео

Комбинация подходов:

Сложные сервисы (следующая лекция)

Агентность: когда она нужна?

Агентность нужна когда:

- Задача требует 3+ шагов с разными инструментами
- Каждый шаг зависит от результата предыдущего
- Невозможно заранее предсказать последовательность шагов
- Нужна адаптация к промежуточным результатам

Пример: "Найди 5 последних статей про квантовые вычисления и создай summary" → поиск → чтение каждой → анализ → суммаризация

Агентность: когда она НЕ нужна?

Агентность НЕ нужна когда:

- Фиксированный pipeline (используйте оркестратор)
- 1-2 шага (используйте Tools напрямую)
- Важна скорость и детерминированность
- Высокие требования к надежности

Пример: "Посчитай 8547 × 37129" → просто Tool (калькулятор)

Принцип: начинайте с простого, усложняйте по необходимости

Подробнее: отдельная лекция по агентным системам

Что дальше?

На следующей лекции мы перейдем к практике:

"Многофункциональный сервис на базе LLM"

- Архитектура комплексных сервисов
- Оркестрация функций и модальностей
- Практическая реализация RAG-системы
- Мониторинг и observability
- Реальные примеры из продакшена

Заключение

Ключевые выводы

1. Stand-alone LLM имеет принципиальные ограничения
2. Инструменты (Tools) решают проблему действий
3. RAG решает проблему актуальности и приватных данных
4. Memory решает проблему контекста
5. Мультимодальность расширяет типы ввода
6. Комбинирование подходов создает мощные сервисы
7. Выбирайте подход под конкретную задачу

Путь расширения LLM

1. Stand-alone LLM
2. + Tools (Function Calling)
3. + RAG (Knowledge Base)
4. + Memory (Personalization)
5. + Multimodal (Rich Input)
6. = Многофункциональный сервис
7. (далее → Агентные системы)

Связь с предыдущими лекциями

1. Лекция 1:

   мы узнали, как работают LLM → сегодня узнали их границы

2. Лекция 2:

   LoRA позволяет адаптировать модель → но не решает проблему актуальности

3. Лекция 3:

   бенчмарки показывают качество → но не покажут ограничения конкретного use case

Вывод: валидация LLM требует понимания контекста применения

Предпросмотр следующей лекции

Лекция 5: "Многофункциональный сервис на базе LLM"

Темы:

- Архитектура production LLM-систем
- Оркестрация инструментов
- Мониторинг и observability
- Примеры реальных систем

Домашнее задание

Практическая работа с Tool Use

- Jupyter Notebook с готовым кодом
- Задача: понять и модифицировать систему с инструментами
- Работа с локальной моделью (Ollama + Llama 3.2)
- Не требуется внешних API
- Можно выполнить в Google Colab

Структура задания

Что в ноутбуке:

1. Часть 1: LLM без инструментов
2. Часть 2: Создание инструмента-калькулятора
3. Часть 3: LLM с инструментами
4. Часть 4: Ваша задача — добавить новую функцию

Что нужно сделать:

- Создать функцию get_length() для подсчета символов
- Добавить описание функции в систему
- Протестировать работу

Критерии оценки

Максимум: 10 баллов

4	3	2	1
Корректность функции	Описание инструмента (JSON Schema)	Интеграция с системой	Тестирование и примеры

Как начать работу

Шаги выполнения:

01

Скачайте ноутбук из материалов курса

02

Откройте в Google Colab

03

Запустите все ячейки по порядку

04

Выполните задание в последней части

05

Сохраните и пришлите в телеграмм @svet_ds с названием ДЗ: "ДЗ_1_Tool_Use_<ФИО>"

Дедлайн: до 28.01.2026 (неделя на выполнение)

Следующая лекция:

"Многофункциональный сервис на базе LLM"

Домашнее задание:

Jupyter Notebook с Tool Use

Вопросы и помощь:

Чат курса