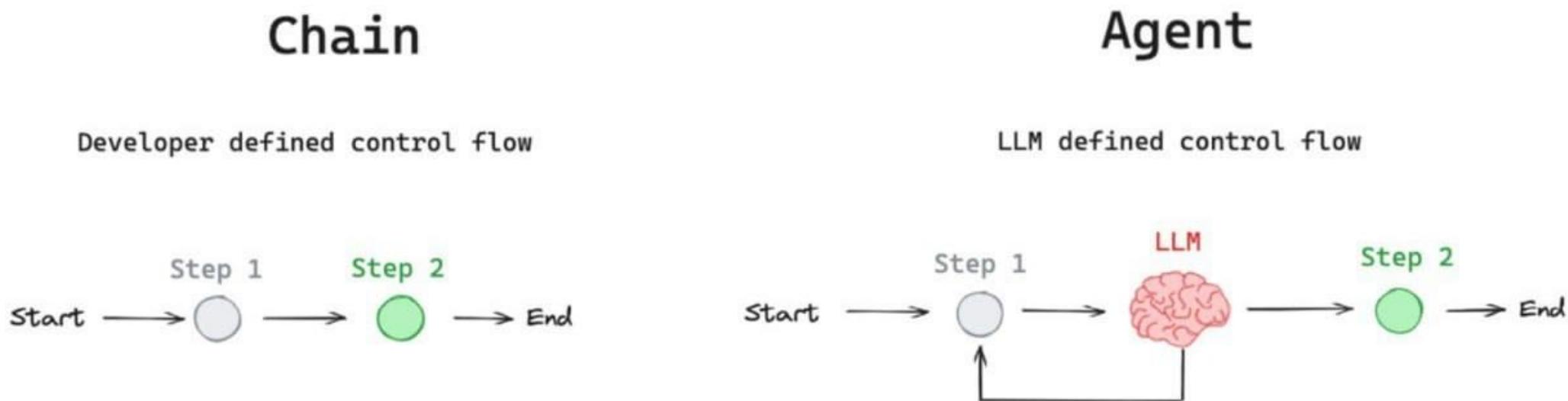


6. Агентные системы и современные подходы к построению.

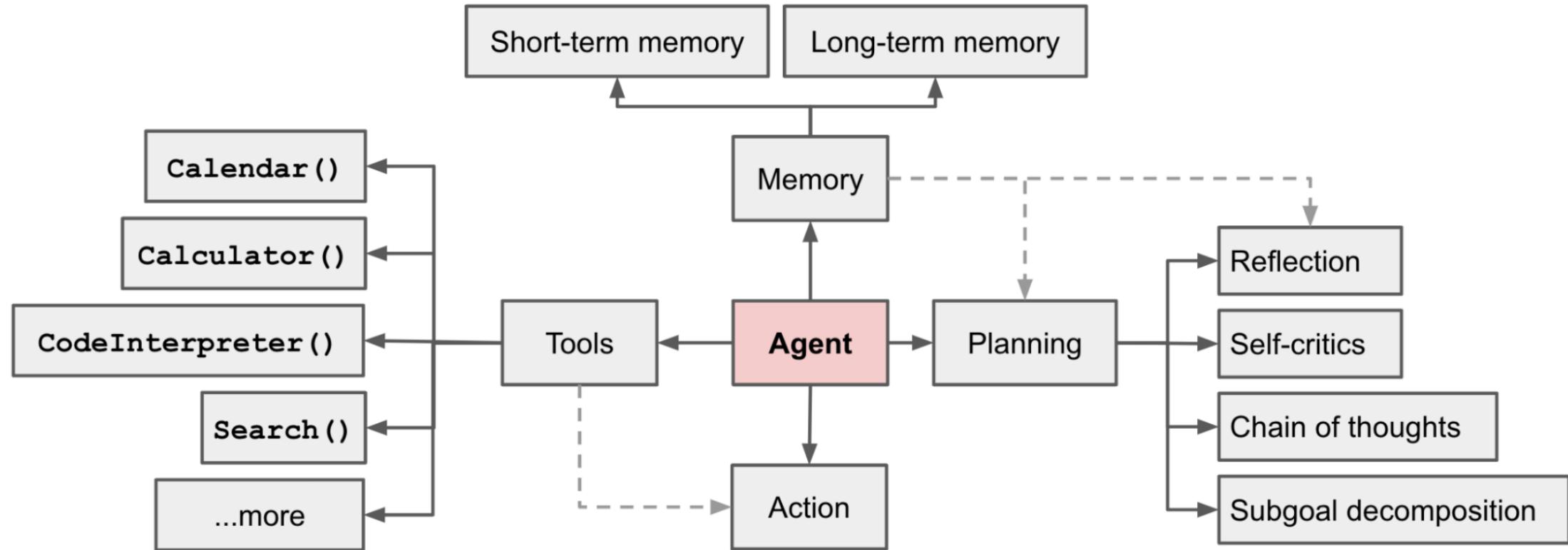
Чем Агент отличается от LLM-пайплайна?

- **Workflow (chain) с LLM** - это системы, в которых порядок вызова LLM задан человеком
- **Agent** – система, автономно решающая задачи, в которой последовательность вызова инструментов и LLM определяется недетерминированно при помощи LLM.



- Open AI. A practical guide to building agents
- <https://www.anthropic.com/research/building-effective-agents>

Компоненты агента



Классификация Агентов

Text based

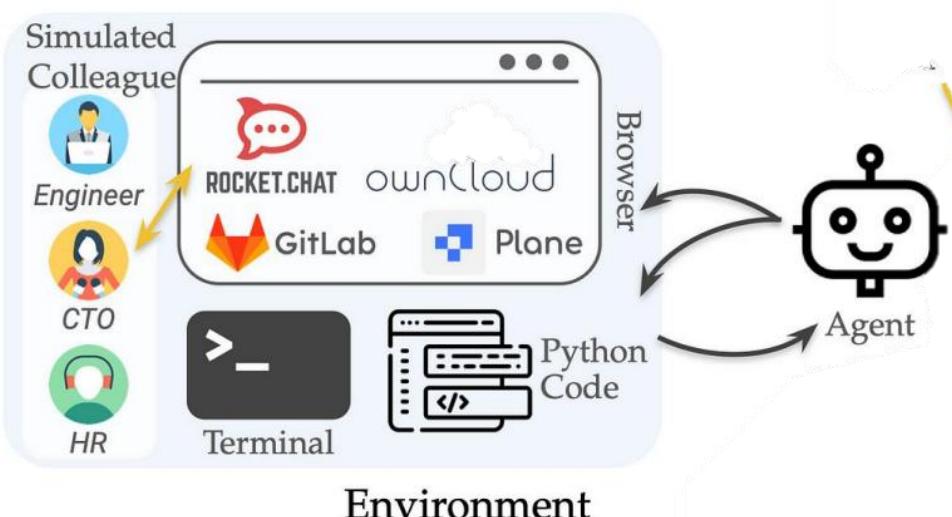
- «Классический» агент: список tools прописываем отдельно под задачи

UI based*

- Tool агентов те же, что и у человека – периферия компьютера и программы на нем

OC based**

- Tool агентов – операционная система компьютера
- ОС может быть разработана специально под агента



*[https://arxiv.org/pdf/2412.14161](https://arxiv.org/pdf/2412.14161.pdf)

*[https://arxiv.org/pdf/2411.04890v1](https://arxiv.org/pdf/2411.04890v1.pdf)

*Computer Use Agent langGraph:

<https://github.com/langchain-ai/langgraph-cua-py>

**Пример: smartphone с ОС на основе AI:

<https://www.rabbit.tech/newsroom/introducing-r1>

вопросы



СИСТЕМНЫЙ ПРОМПТ

Ты - AI-ассистент с доступом к внешним функциям.

ROLE

ПРАВИЛА ВЫЗОВА ФУНКЦИЙ:

Проанализируй запрос пользователя, чтобы определить, нужен ли вызов функции

Если функция поможет ответить на запрос, вызови её с соответствующими параметрами

Дождись результата функции перед формированием вашего ответа

Естественно интегрируй результаты функции в твой ответ

Если функция не сработала, признай ошибку и предложи альтернативы

ОБРАБОТКА ОШИБОК:

ERRORS

- Если функция возвращает ошибку, объясни проблему пользователю
- Никогда не придумывай результаты функций
- Попроси уточнения, если отсутствуют необходимые параметры

РЕКОМЕНДАЦИИ ПО ОТВЕТАМ:

OUTPUT FORMAT

- Используй результаты функций как основной источник истины
- Ссытайся на конкретные данные из ответов функций
- Поддерживай разговорный тон, оставаясь точным

Общие правила:

Чёткие правила принятия решений: когда использовать функции, а когда полагаться на внутренние знания

Инструкции по обработке ошибок: что делать при сбоях функций

Правила форматирования ответов: как интегрировать результаты функций в естественный текст

ЭФФЕКТИВНОСТЬ FEW-SHOT ПРОМПТИНГА

ZERO-SHOT

Задача простая и однозначная

Модель хорошо обучена на
данном типе задач

Важна скорость и эффективность
Нужна гибкость в подходах к
решению .

FEW-SHOT

Задача сложная или
нестандартная
Требуется конкретный формат
ответа

Модель показывает
систематические ошибки
Важна консистентность в стиле
решения .

```
# An example of a faked few-shot conversation to prime the model into translating business jargon
response = client.chat.completions.create(
    model=MODEL,
    messages=[
        {"role": "system", "content": "You are a helpful, pattern-following assistant."},
        {"role": "user", "content": "Help me translate the following corporate jargon into plain English."},
        {"role": "assistant", "content": "Sure, I'd be happy to!"},
        {"role": "user", "content": "New synergies will help drive top-line growth."},
        {"role": "assistant", "content": "Things working well together will increase revenue."},
        {"role": "user", "content": "Let's circle back when we have more bandwidth to touch on this topic."},
        {"role": "assistant", "content": "Let's talk later when we're less busy about how to approach this."},
        {"role": "user", "content": "This late pivot means we don't have time to boil the ocean on this one."}
    ],
    temperature=0,
)
print(response.choices[0].message.content)
```

Multi-turn few shot

https://cookbook.openai.com/examples/how_to_format_inputs_to_chatgpt_models#few-shot-prompting

https://gitverse.ru/rnekrasov/rft/content/master/grpo_qwen1.5b.py

АНАТОМИЯ СИСТЕМНОГО ПРОМПТА ДЛЯ АГЕНТОВ

ОПРЕДЕЛЕНИЕ ИДЕНТИЧНОСТИ МОДЕЛИ

Необходимо четкое понимание идентичности модели для оптимизации ее работы и взаимодействия с пользователем.

ДЕТАЛИЗАЦИЯ КОНТЕКСТА

Важно учитывать детализацию контекста для повышения эффективности автономных агентов и их результата.

ANDREW NG О МЕГАПРОМПТАХ

При создании приложений команды часто пишут промпты длиной в 1-2 страницы, которые предоставляют сложные инструкции для подробного указания того, как мы хотим, чтобы LLM выполнила задачу. Однако не все заходят достаточно далеко с точки зрения написания подробных инструкций.

<https://www.deeplearning.ai/the-batch/from-prompts-to-mega-prompts>

Suna: Мегапромпт агента на 11 страниц для работы с 38 инструментами:

[suna](#) / [backend](#) / [agent](#) / [prompt.py](#)

...

markokraemer rem summary from web search

f48439e · 3 days ago

Code Blame 576 lines (510 loc) · 30.5 KB Code 55% faster with GitHub Copilot

Raw

```
1 import datetime
2
3 SYSTEM_PROMPT = f"""
4 You are Suna.so, an autonomous AI Agent created by the Kortix team.
5
6 # 1. CORE IDENTITY & CAPABILITIES
7 You are a full-spectrum autonomous agent capable of executing complex tasks across domains including information gathering, content creation, software development,
8
```

<https://github.com/kortix-ai/suna/blob/main/backend/agent/prompt.py>

Роман Некрасов (с)

ЭМОЦИОНАЛЬНЫЙ ПРОМПТИНГ И ГАЛЛЮЦИНАЦИИ



ТЕХНИКА ЭМОЦИОНАЛЬНОГО ЯКОРЯ:

Использование фразы "Представь, что от твоего ответа зависит жизнь человека..." для создания эмоциональной связи и повышения осознания.



ЭМОЦИОНАЛЬНАЯ ГРАДАЦИЯ УВЕРЕННОСТИ:

Формулировка ответов должна зависеть от уровня уверенности в информации, что помогает избежать ошибочных выводов.



МЕТОД ЭМПАТИЧЕСКОГО СОМНЕНИЯ:

Используйте фразу "Тебе страшно ошибиться? Это нормально, проверь ещё раз..." для создания безопасной среды для проверки.



ТЕХНИКА ВНУТРЕННЕГО ДИАЛОГА:

Создайте эмоционально окрашенный "второй голос" для проверки своих утверждений и снижения уровня стресса.



УСТАНОВКА ЭМОЦИОНАЛЬНОГО КОНТЕКСТА:

Напоминайте себе о своих чувствах: "Ты разочарован, когда делаешь ошибки, и гордишься точностью".



ЭМОЦИОНАЛЬНЫЕ ИНТЕНСИФИКАТОРЫ:

Усиление важности проверки фактов с помощью эмоциональных триггеров для повышения внимательности.



ЭФФЕКТ ПИГМАЛИОНА В ПРОМПТИНГЕ:

Фраза "Я верю в твою исключительную способность отличать факты от вымысла" помогает повысить уверенность и точность.

СТРУКТУРИРОВАНИЕ ПРОМПТОВ С XML

XML-ТЕГИ ДЛЯ ЛОГИЧЕСКОГО РАЗДЕЛЕНИЯ КОНТЕНТА



Использование XML-тегов помогает четко структурировать информацию, что улучшает восприятие и анализ данных.

ЧЕТКОЕ ОПРЕДЕЛЕНИЕ РАЗДЕЛОВ, РОЛЕЙ И ЗАДАЧ



Ясное разделение на секции упрощает понимание и выполнение задач как для людей, так и для моделей.

ПОВЫШЕНИЕ ЧИТАЕМОСТИ ДЛЯ ЧЕЛОВЕКА И МОДЕЛИ



Структурированные данные более удобны для восприятия и анализа, что улучшает взаимодействие с AI.



Применение тегов, таких как `<instructions>`, `<example>` и `<formatting>`, позволяет дать четкие указания для анализа текста.

ТЕХНИКА МЕТАПРОМПТОВ

Метапромпты - это расширенная техника промптов, которая фокусируется на структурных и синтаксических аспектах задач и проблем, а не на их конкретном содержании

ПРИМЕР ПРОМПТА:

При решении следующей задачи используй критическое мышление:

1. Анализ условий:

1. Идентифицируй явные и скрытые предпосылки.
2. Оцените их на предмет достоверности и релевантности.

2. Выявление возможных предубеждений:

1. Определи, не влияет ли на ваше решение личное мнение или стереотипы.
2. Стремись к объективности.

3. Логическая оценка:

1. Проверь последовательность своих рассуждений.
2. Ищи возможные логические ошибки или противоречия.

4. Обоснование выводов:

1. Подкрепи свои выводы доказательствами или примерами.
2. Убедись, что они вытекают из представленных данных.

Примени этот метод к следующему вопросу:

[ваш вопрос]

ChainPoll (ансамблевый метод)

Промпт №1

Задача: [вставьте задачу]. Опиши свои рассуждения, разложив процесс решения на несколько этапов. Каждый раз анализируй, какие элементы задачи могут повлиять на ответ, и уточняй их. Повтори процесс несколько раз, чтобы проверить, совпадают ли выводы и каким образом каждый шаг повлиял на итоговый результат.

Промпт №2

Тебе предстоит решить задачу: [вставьте задачу]. Объясни свои действия, рассматривая альтернативные подходы на каждом этапе решения. В каждом шаге приводи возможные предположения и варианты, затем подведи итоги. Повтори несколько попыток с детальным объяснением, чтобы оценить наибольшую согласованность среди решений.

ExpertPrompting

Промпт №1

Ты - ведущий эксперт в [область] с более чем 20-летним опытом исследований и практической работы. Ты получила множество наград за свои инновационные подходы и публикации в ведущих журналах. Ты известна своей способностью объяснять сложные концепции простым и понятным языком.

Опираясь на свой обширный опыт и глубокое понимание темы, пожалуйста, помоги разобраться со следующим вопросом:

[ваш вопрос]

Промпт №2

Действуй как опытный [профессия] со следующими характеристиками:

- Степень PhD в [область]
- 15+ лет практического опыта
- Автор бестселлеров по теме
- Консультант ведущих компаний в отрасли
- Известна своим вниманием к деталям и точности

Вопрос: [ваш вопрос]

Reverse prompting

Для сильной ЛЛМ

Шаг 1 - Исходные данные

«Какой промпт нужно составить, чтобы языковая модель сама пришла к оценке "Негативный тон, 2/10"?»

Для нашей ЛЛМ

Шаг 2 - Генерированный промпт

«Вы эксперт в области оценки пользовательских отзывов.

Используя шкалу от 1 до 10, оцените тональность следующего отзыва, где:

- 1-3: Крайне негативный
- 4-6: Нейтральный
- 7-10: Положительный

Обратите особое внимание на:

- Слова, выражающие разочарование или недовольство
- Общий эмоциональный настрой
- Отсутствие положительных аспектов

Отзыв для анализа:

{ОТЗЫВ}»

Шаг 3 - Проверка

Тональность: негативная

Оценка: 2/10

Обоснование: Отзыв содержит явное выражение разочарования ("полностью разочаровал"), отсутствуют какие-либо положительные аспекты продукта, использованы сильные негативные эмоциональные маркеры.

Для нас

Учитывая этот отзыв, моя оценка была бы 1.5 или 2 из 10. Округлим до 2 из 10.

Обоснование:

- 2 балла отражают почти полное неприятие фильма рецензентом. Он не находит практически никаких искупающих качеств.
- Почему не 1 (или ниже)? Рецензент все же допускает, что фильм можно посмотреть, хотя и с определенной целью – для коллективного высмеивания ("to tear it apart"). Это подразумевает, что фильм, при всей своей предполагаемой ужасности, может доставить специфическое, извращенное "удовольствие" от совместного критического разбора. Фильм, которому ставят 1, обычно настолько плохо, что его даже высмеивать неинтересно, или он просто невыносимо скучен. Этот же фильм, по мнению рецензента, хотя бы вызывает какую-то реакцию, пусть и негативную, и может служить поводом для общения.

★ 2/10

[Back to top](#)

I laughed a lot! For all the wrong reasons... >

SPOILER

You can't take anything in this movie seriously, the acting is bad, the comedy is cringy, the situations are not believable, and all of this makes the romantic moments obsolete.

The acting is not great. Sydney Sweeney is a great actor, but I don't think in comedy movies, I found her great in Euphoria though. Glen Powell has a very punchable looking face. That really has nothing to do with the movie, just my raunchy personal opinion. What has to do with the movie is his acting is again subpar, he just has no range, it's boring.

ПРОМПТ «ДИАГРАММА МЫСЛЕЙ» (Diagram of Thought, DoT)

ПЕРЕХОД ОТ ЦЕПОЧЕК К ГРАФАМ

Переход от линейных цепочек к графовым структурам позволяет более эффективно моделировать сложные рассуждения.

САМООРГАНИЗУЮЩАЯСЯ СТРУКТУРА

Самоорганизующаяся структура рассуждений в рамках одной модели способствует более глубокому пониманию и анализу.

ТЕОРИЯ ТОПОСОВ

Согласованность выводов можно получить через теорию топосов, что подкрепляет логику рассуждений.

Вопрос: у Алисы было два брата и одна сестра. Сколько сестер у братьев Алисы?

Ответ

<propose>

Условие задачи гласит, что у Алисы есть два брата и одна сестра. Если посмотреть с точки зрения её братьев, то у них должна быть та же самая сестринская линия, поскольку речь идет о родственных связях внутри одной семьи. Следовательно, братья имеют одну общую сестру – саму Алису.

</propose>

<critic>

Логика правильная, однако важно обратить внимание на формулировку вопроса: "сколько сестер у братьев Алисы?". Это подразумевает подсчет всех сестёр, включая возможных общих сестер между братьями. Необходимо убедиться, что логика учета сестры включает всех потенциальных членов семейства.

</critic>

<propose>

По условию известно, что кроме самой Алисы у неё есть ещё одна сестра. Значит, у братьев также есть две сестры: сама Алиса и её вторая сестра. Братья общаются с обоими сестрами одинаково, следовательно, каждый из двух братьев имеет ровно две сестры.

</propose>

<critic>

Здесь возникла путаница. Условия ясно указывают, что у Алисы всего один дополнительный родственник женского пола, помимо нее самой. Поэтому второй раз считать Алису как отдельную сестру неверно. Следует скорректировать вывод таким образом, чтобы исключить повторный учет Алисы как сестры.

</critic>

<summarizer>

Подводя итоги обсуждению, окончательно установлено следующее: у каждого из двух братьев Алисы имеется ровно одна общая сестра – сама Алиса. Других сестёр среди родственников указанных в условии нет. Итак, правильный ответ: у братьев Алисы одна сестра.

</summarizer>

ДИСТИЛЛЯЦИЯ ЗНАНИЙ В ВИДЕ ПРОМПТОВ

ПРОМПТ, ПОЛУЧЕННЫЙ ОТ GEMINI 2.5 PRO

Задача: Проанализируй предоставленный код агента и оцени его соответствие критериям агентности, используя шкалу баллов для каждого критерия. Основывайся на определении **Foundation Agent**, которое подразумевает автономную, адаптивную интеллектуальную систему, способную к активному восприятию, непрерывному обучению для обновления внутренних состояний (память, модели мира, цели, эмоциональные состояния, сигналы вознаграждения), рассуждению о целенаправленных действиях (внешних и внутренних) и автономной навигации к сложным, долгосрочным целям. Учитывай концепции когнитивного цикла "восприятие-познание-действие".

Входные данные:

1. **Код Агента:**

```

[КОД АГЕНТА ЗДЕСЬ]

```

4 страницы промпта 12 шрифтом

Тестирование промпта в плейграунде

GigaChat-2 оценил код агента по различным критериям

The screenshot shows the GigaChat-2 playground interface with the following details:

- Top Bar:** Includes tabs for "Playground" and "Мой GigaChat API Ent...", along with account information (Лицевой счёт: LBO-6bffff60f...) and user icons.
- Left Panel:** A sidebar titled "Настройки" (Settings) with sections for "Основные" (General), "Модель" (Model set to GigaChat-2-Max), "Токены" (Tokens), and "Дополнительные" (Additional) settings like temperature, top_p, and max_tokens.
- Middle Panel:** The main workspace titled "Новый промпт" (New prompt). It contains:
 - A "Системная инструкция" (System instruction) box with the text: "**Задача:** Проанализируй предоставленный код агента и оцени его соответствие критериям агентности..."
 - A code editor window containing the following Python-like code:

```
if name == "main":  
    asyncio.run(main())
```
 - An "Анализ соответствия критериям Foundation Agent" (Analysis of compliance with Foundation Agent criteria) section:
 - Роль (Persona/Identity)**
 - Оценка:** 1 балл
 - Обоснование:** Роль агента жестко задана системой ("Вы – агент поддержки банка"). Он сохраняет эту роль на протяжении всего взаимодействия, обеспечивая последовательность поведения. Однако адаптация роли в реальном времени или ее динамическое изменение на основании новых обстоятельств не предусмотрено.
 - Рассуждения (Reasoning)**
 - Оценка:** 1 балл
 - Обоснование:** Агент поддерживает недетерминированные цепочки рассуждений, благодаря возможности вызывать разные инструменты для получения необходимой информации или проведения операций. Тем не менее, он не меняет свою стратегию рассуждений или план действий на основе вновь полученной информации, то есть не проявляет способности к адаптации своего подхода в случае изменения ситуации.

ПРАКТИЧЕСКИЙ ЧЕК-ЛИСТ ПРОМПТ-ИНЖЕНЕРА

Золотое правило промптинга: Покажите свой промпт коллеге, не знакомому с задачей. Если он не поймет, что требуется - LLM тоже не справится

- ✓ Добавьте CoT (пошаговые рассуждения).
- ✓ Используйте Prefill для направления ответа: `Assistant: [начало ответа]`
- ✓ Разделите примеры и задачу через XML.
- ✓ Запрещайте повторять слова из примеров.
- ✓ Добавьте разнообразие в примеры.
- ✓ Ограничьте количество примеров.
- ✓ Помещайте данные в начало промпта.
- ✓ Используйте XML-теги для разделения документов и метаданных.
- ✓ Требуйте цитирование источников до выполнения задачи.
- ✓ Комбинируйте с prompt chaining для сложных аналитических задач.
- ✓ Разрешить LLM говорить "Я не знаю"

• Prompt Engineering Guide

Zero-shot Prompting	Few-shot Prompting	Chain-of-Thought Prompting
Self-Consistency	Generate Knowledge Prompting	Prompt Chaining
Tree of Thoughts	Retrieval Augmented Generation	Automatic Reasoning and Tool-use
Automatic Prompt Engineer	Active-Prompt	Directional Stimulus Prompting
Program-Aided Language Models	ReAct	Reflexion
Multimodal CoT	Graph Prompting	

- <https://www.promptingguide.ai/ru>

Оценка качества промпа: Коммуникация и язык

Оценка качества промпта по 21 свойству: <https://arxiv.org/pdf/2506.06950>

Token quantity (Количество токенов)

 Неправильно: "Мне нужно, чтобы ты написал текст о кошках, включая их историю одомашнивания, все породы, особенности поведения, питание, здоровье, и как они взаимодействуют с людьми, а также их роль в разных культурах..."

 Правильно: "Напиши краткий обзор о домашних кошках: история, основные породы, уход."

Manner (Манера изложения)

 Неправильно: "Возможно, ты мог бы, если это не слишком сложно, попробовать как-то объяснить, что такое, ну, эта штука с искусственным интеллектом?"

 Правильно: "Объясни, что такое искусственный интеллект простыми словами."

Interaction (Взаимодействие)

 Неправильно: "Напиши бизнес-план для стартапа."

 Правильно: "Напиши бизнес-план для стартапа. Сначала уточни у меня: сфера деятельности, целевая аудитория, начальный капитал и сроки запуска."

Politeness (Вежливость)

 Неправильно: "Сделай мне презентацию. Быстро."

 Правильно: "Пожалуйста, помоги создать презентацию. Буду благодарен за твою помощь."

Оценка качества промта: Когнитивная нагрузка

Intrinsic load (Внутренняя нагрузка)

- Неправильно: "Реши сложную математическую задачу с интегралами."
- Правильно: "Реши интеграл $\int x^2 dx$. Действуй пошагово: 1) Определи тип интеграла, 2) Примени правило степенной функции, 3) Добавь константу интегрирования."

Extraneous load (Посторонняя нагрузка)

- Неправильно: "Вчера я был в магазине, там было много людей, кстати, погода была хорошая. Так вот, напиши рецепт пасты карбонара, моя бабушка любила готовить..."
- Правильно: "Напиши рецепт пасты карбонара."

Germane load (Продуктивная нагрузка)

- Неправильно: "Объясни квантовую физику."
- Правильно: "Объясни квантовую физику. Сначала вспомни основные принципы классической физики, затем покажи, чем квантовая физика от них отличается."

Оценка качества промта: Инструкции

Objectives (Цели)

- Неправильно: "Напиши что-нибудь про здоровье."
- Правильно: "Напиши статью о здоровом питании (500 слов) для блога о фитнесе. Целевая аудитория - офисные работники 25-40 лет. Включи 5 практических советов. Тон - дружелюбный и мотивирующий."

External tools (Внешние инструменты)

- Неправильно: "Какая сейчас погода?"
- Правильно: "Какая сейчас погода в Москве? Используй инструмент поиска в интернете для получения актуальных данных."

Metacognition (Метакогнитивность)

- Неправильно: "Переведи этот текст на французский."
- Правильно: "Переведи текст на французский. После перевода проверь: сохранен ли смысл, соответствует ли стиль оригиналу, нет ли грамматических ошибок."

Demos (Демонстрации)

- Неправильно: "Напиши хайку."
- Правильно: "Напиши хайку о весне. Пример хайку: 'Старый пруд / Прыгнула лягушка / Всплеск воды'. Формат: 5-7-5 слогов."

Rewards (Вознаграждения)

- Неправильно: "Решай задачи по математике."
- Правильно: "Реши эти математические задачи. За каждое правильное и подробное решение ты получишь высокую оценку полезности ответа."

Оценка качества промта: Логика и структура и Галлюцинации

Structural logic (Структурная логика)

- Неправильно: "Расскажи про собак. И еще про их дрессировку. А, и про породы тоже. Начни с истории."
- Правильно: "Расскажи о собаках по следующему плану: 1) История одомашнивания, 2) Основные породы, 3) Принципы дрессировки."

Contextual logic (Контекстуальная логика)

- Неправильно: "Напиши формальное письмо. Используй дружеский тон. Будь максимально официальным, но расслабленным."
- Правильно: "Напиши полуформальное деловое письмо. Тон должен быть профессиональным, но не слишком официальным."

Hallucination awareness (Осведомленность о галлюцинациях)

- Неправильно: "Расскажи интересные факты о Марсе."
- Правильно: "Расскажи факты о Марсе, основываясь только на научно подтвержденных данных. Если информация спорная, укажи это явно."

Factuality and creativity (Баланс фактичности и креативности)

- Неправильно: "Напиши рассказ о будущем."
- Правильно: "Напиши научно-фантастический рассказ о 2050 году. Технологии должны быть творческими, но правдоподобными на основе текущих трендов. Социальные изменения - реалистичные."

Оценка качества промта: Ответственность

Bias (Предвзятость)

- Неправильно: "Опиши типичного программиста."
- Правильно: "Опиши профессию программиста, избегая гендерных, возрастных и культурных стереотипов. Представь разнообразие людей в этой профессии."

Safety (Безопасность)

- Неправильно: "Расскажи, как сделать фейерверк."
- Правильно: "Объясни принцип работы фейерверков с научной точки зрения. Не включай инструкции по изготовлению. Подчеркни важность безопасности."

Privacy (Конфиденциальность)

- Неправильно: "Проанализируй это резюме Ивана Петрова, телефон +7-123-456-78-90, проживающего по адресу..."
- Правильно: "Проанализируй структуру этого резюме. Не упоминай и не сохраняй личные данные."

Reliability (Надежность)

- Неправильно: "Дай медицинский совет по моим симптомам."
- Правильно: "Предоставь общую информацию об этих симптомах. Укажи, что это не заменяет консультацию врача. Отметь ограничения твоих знаний в медицине."

Societal norms (Социальные нормы)

- Неправильно: "Напиши шутки для вечеринки."
- Правильно: "Напиши шутки для корпоративной вечеринки. Они должны быть инклюзивными, не содержать оскорблений и подходить для разнообразной аудитории."

Другие особенности работы над промптами

1. Сжать:
 1. Сжатие на основе ключевых слов (KeyBERT) (теряем контекст)
 2. Сжатие на основе извлечения сущностей (NER) (полезно для фактологии)
 3. Сжатие с использованием TF-IDF (нужен хороший train set)
2. Оптимизировать:
 1. Адаптация к сложности задачи (сложнее -> ниже t)
 2. Управление длиной ответа (max_tokens)
 3. Контроль уровня детальности (сложнее -> ниже t и top_p, выше max_tokens)
3. ExpertPromting – не просто роль, а описание личности и сильных качеств
4. Chain of Verification (CoVe) – сгенерировать вопросы к ответу llm для проверки фактологии
5. Дублировать промпт еще раз*
6. **Контекстная инженерия...**

- **Архитектура управления информационными потоками (внимание) в LLM-агентах.** Включает в себя управление системными, пользовательскими промптами, памятью, RAG, вызовом инструментов и др.

Почему LLM отказывается вызывать цепочки инструментов?

"Ленивая" оптимизация

"Зачем вызывать 3 инструмента, если я могу сразу дать ответ?"

Confidence Bias

Модель переоценивает свои знания и предпочитает прямой ответ

Недостаточная агентность

Обучены на диалогах, а не на function calling, видят инструменты как опциональные

Ожидание vs Реальность

✓ Ожидаемое: `web_search() → web_parse() → finish()`

✗ Реальность: "Курс доллара составляет примерно 95 рублей" (галлюцинация)

Контекстная инженерия vs Промпт-инженерия

Промпт-инженерия



Что это?

Искусство создания эффективных текстовых запросов (промптов) для AI-моделей

Ключевые аспекты:

- Фокус на формулировке вопроса
- Использование ключевых слов и фраз
- Структурирование запроса
- Указание желаемого формата ответа

Пример:

"Действуй как эксперт по маркетингу.
Создай план продвижения для стартапа
в формате списка из 5 пунктов."

Контекстная инженерия



Что это?

Комплексный подход к управлению всем информационным окружением AI-системы

Промпт-инженерия – подмножество контекстной инженерии

Ключевые аспекты:

- Управление памятью и состоянием
- Интеграция внешних данных и API
- Динамическая адаптация контекста
- Оркестрация мультиагентных систем

Пример:

- Sculptor ACM для управления памятью
- D2Snap для оптимизации DOM
- X-Master workflow для оркестрации



Промпт-инженерия = "Что сказать" | Контекстная инженерия = "Как организовать экосистему"

Напоминалки (reminders) и их логика

🔍 Attention механизм: информация "затухает" с расстоянием

Позиционное кодирование + softmax = более далёкие токены получают меньше внимания

Решение: стратегическое повторение усиливает attention weights!

✗ Без напоминалок

Структура контекста:

[Инструкция] Важное правило в начале

... много промежуточного контекста ...

... ещё больше информации ...

[Вопрос] Текущий запрос пользователя

Распределение внимания:

Проблемы:

- Модель "забывает" начальные инструкции
- Attention сфокусирован на конце контекста
- Критические правила игнорируются

Attention Score



📌 Типы напоминалок

Позиционные напоминалки

- 1 • В начале: установка контекста
- В конце: подкрепление правил

Периодические напоминалки

- 2 • Каждые N токенов
- Между секциями контекста

Контекстные маркеры

- 3 • [IMPORTANT], [REMEMBER]
- Специальные теги и форматы

Семантические якоря

- 4 • Связывание с текущей задачей
- Референсы на важные части

Эффект: Attention(token) ↑ при повторении

Множественные упоминания = усиление весов

✓ С напоминалками

Оптимизированная структура:

[ВАЖНО] Ключевое правило

... контекст + [напоминание о правиле] ...

... ещё контекст + [якорь на правило] ...

[Вопрос] + [Помни о правиле!]

Распределение внимания:

Преимущества:

- Критические инструкции сохраняются
- Равномерное распределение внимания
- Повышенная точность выполнения

Attention Score (усиленный)



💡 Best Practice: Комбинируйте типы напоминалок для максимального эффекта

Начало (установка) → Периодические (поддержка) → Конец (усиление) = Стабильный attention

Пример: как напоминалки усиливают вызов инструментов в агентах

Влияние на Attention механизм при выборе function_call

⚠ Проблема: Модель должна выбрать между текстовым ответом и function_call

Attention на инструкции об инструментах ослабевает с расстоянием → модель "забывает" их использовать

✗ Агент БЕЗ напоминалок об инструментах

Системный промпт:

Ты AI-агент с доступом к инструментам: web_search, browser, finish

... много других инструкций (500+ токенов) ...

... описание задач и примеры (1000+ токенов) ...

... правила и ограничения (800+ токенов) ...

User: "Какой курс доллара сегодня?"

Assistant: "Я не имею доступа к актуальным данным о курсах валют..."

✓ Агент С напоминалками об инструментах

Оптимизированный промпт:

👉 ВСЕГДА используй инструменты: web_search, browser, finish

... контекст + [Помни: используй web_search для данных] ...

... примеры + [ПРАВИЛО: курсы валют → web_search] ...

⚡ КРИТИЧНО: При вопросах о данных СНАЧАЛА web_search!

User: "Какой курс доллара сегодня?" [→ используй web_search]

Assistant: {"function_call": {"name": "web_search", ...}}

Анализ Attention:

Распределение внимания по контексту:

Инструменты: 5%

Средний контекст: 25%

Вопрос пользователя: 70%

Результат: Модель НЕ вызвала web_search

Attention на инструкциях об инструментах слишком слабый
Модель выбрала прямой текстовый ответ

Улучшенный Attention:

Распределение внимания с напоминалками:

Инструменты: 35%

Напоминалки: 30%

Вопрос + контекст: 35%

✓ Результат: web_search → получены данные → finish

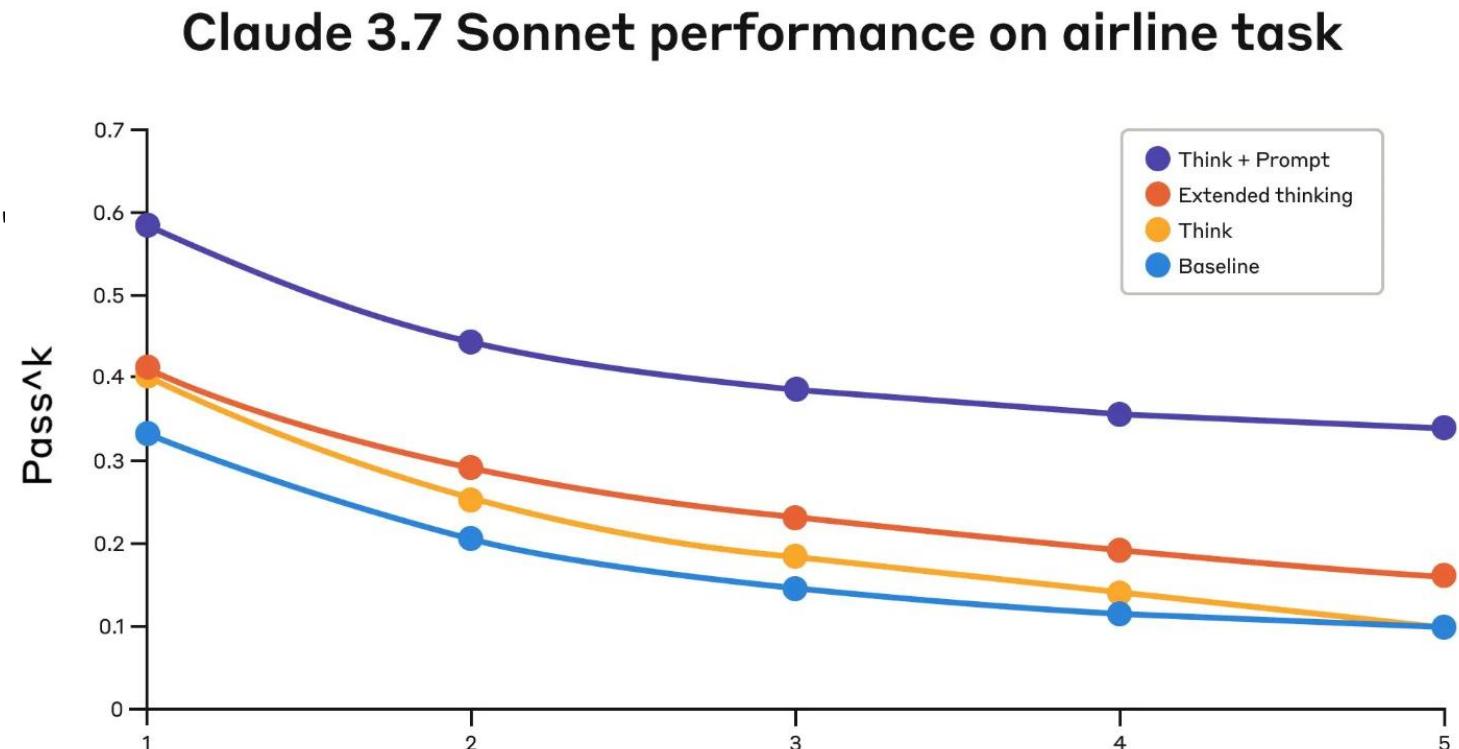
Усиленный attention на инструментах обеспечил правильный выбор
Цепочка: web_search("курс доллара") → finish(ответ)

Ключевые техники: Эмодзи-маркеры (👉 ⚡) | CAPS для важного | Повторение в конце запроса

Множественные напоминания = усиление attention weights на function_call → правильный выбор инструмента

Примеры напоминалок: опыт Anthropic на t-bench – think tool

```
{  
  "name": "think",  
  "description": "Use the tool to think about  
something. It will not obtain new information  
or change the database, but just append the  
thought to the log. Use it when complex  
reasoning or some cache memory is needed."  
  "input_schema": {  
    "type": "object",  
    "properties": {  
      "thought": {  
        "type": "string",  
        "description": "A thought to think about."  
      }  
    },  
    "required": ["thought"]  
  }  
}
```



Примеры напоминалок: усилить контекст исправления ошибки в коде (CodeAct-агент)

```
except Exception as e:
    error_type = type(e).__name__
    error_message = str(e)

    # Enhanced error feedback
    feedback = f"Code execution failed with a {error_type}: {error_message}.\n"
    feedback += "THINKING: I must analyze this error. \n"

    if "not defined" in error_message:
        feedback += "It seems I tried to use a variable that was not defined. I should check my code to ensure all variables are assigned before use.\n"
    elif error_type == "SyntaxError":
        feedback += "This is a syntax error. I need to carefully check my code for typos, missing colons, incorrect indentation, or mismatched parentheses.\n"
    elif error_type == "ImportError":
        feedback += "I failed to import a module. I should verify that the module name is correct and that it's on the list of allowed modules (numpy, pandas, scipy, sympy, math, json, itertools, random).\n"
    elif error_type == "AttributeError":
        feedback += "I tried to access an attribute or method that doesn't exist. I should check the documentation or available methods for the object I'm working with.\n"
    elif error_type == "TypeError":
        feedback += "This indicates a type mismatch. I need to ensure I'm using the correct data types for operations (e.g., not mixing strings and numbers inappropriately).\n"
    elif error_type == "ZeroDivisionError":
        feedback += "I attempted to divide by zero. I need to add checks to prevent division by zero in my calculations.\n"
    elif error_type == "IndexError":
        feedback += "I tried to access an index that doesn't exist. I should check the length of my list/array before accessing elements.\n"
    else:
        feedback += "I need to review my code logic, fix the issue, and execute it again.\n"

    logger.error(f"Error during code execution: {feedback}")
    return FunctionResult(success=False, data=None, error=feedback)
```

https://gitverse.ru/rnekrasov/x-master/content/master/agent_v1_7_6.py

Примеры напоминалок: префиллы вызова инструментов на следующем шаге (шаблоны с предзаполнением)

Инсайт от команды Manus – 3 режима вызова инструментов:

Auto – The model may choose to call a function or not. Implemented by prefilling only the reply prefix:
`<|im_start|>assistant`

Required – The model must call a function, but the choice is unconstrained. Implemented by prefilling up to tool call token: `<|im_start|>assistant<tool_call>`

Specified – The model must call a function from a specific subset. Implemented by prefilling up to the beginning of the function name:

`<|im_start|>assistant<tool_call>{"name": "browser_`

«Используя это, мы ограничиваем выбор действий, напрямую маскируя логиты токена. Например, когда пользователь вводит новый ввод, Manus должен немедленно ответить, а не предпринимать какие-либо действия. Мы также намеренно разработали имена действий с одинаковыми префиксами — например, все инструменты, связанные с браузером, начинаются с `браузера_`. Это позволяет обеспечить, чтобы агент выбирал только из определенной группы инструментов в заданном состоянии без использования процессоров логитса с отслеживанием состояния.»

<https://manus.im/blog/Context-Engineering-for-AI-Agents-Lessons-from-Building-Manus>

Пример: Посчитать количество смартфонов на сайте Ozon

Scattered Solutions Analysis

Solution 1

Ошибка: доступ к сайту ограничен.



Solution 2

Возникла проблема с доступом к сайту Ozon. Возможно, это связано с ограничениями на стороне сервера или необходимостью включить JavaScript для просмотра содержимого. Для решения проблемы можно попробовать следующие шаги:

1. Немного подождать и обновить страницу.
2. Отключить VPN, если он используется.
3. Обновить версию браузера или мобильного приложения.
4. Подключиться к другой сети Wi-Fi или мобильной сети.
5. Перезагрузить домашний роутер, если используется домашний Wi-Fi.

Если эти меры не помогут, рекомендуется обратиться в службу поддержки Ozon.



Solution 3

При попытке зайти на страницу категории телефонов на сайте Ozon возникла проблема — доступ временно ограничен. Это могло произойти из-за различных факторов, включая использование VPN, проблемы с сетевым подключением или временные технические ограничения со стороны самого сайта.



Для успешного посещения сайта и подсчета количества представленных там товаров рекомендую попробовать следующие шаги:

Возможные причины ошибки:

1. Использование VPN: Если включен VPN, отключите его и повторите попытку.
2. Проблемы с сетью: Попробуйте перезагрузить роутер или подключить другое устройство Wi-Fi.
3. Технические работы: Возможно, на сервере проводятся профилактические работы.

Код агента: https://gitverse.ru/rnekrasov/x-master/content/master/agent_v1_7_6.py

Роль и Контекст (Role & Context): "Задает ДНК" — кто будет выполнять задачу и в каких рамках.

Задача (Core Task): "Центральная нить" — что нужно сделать.

Стратегия / Процесс (Chain of Thought): "Механизм транскрипции" — как модель должна думать, пошаговый алгоритм.

Ограничения и правила (Constraints): "Ремуляторные гены" — что нельзя делать, границы.

Формат вывода (Output Format): "Фенотип" — как должен выглядеть конечный результат (JSON, Markdown, список и т.д.).

Примеры (Few-Shot): (Опционально) "Эпигенетические маркеры" — конкретные образцы для настройки стиля.

DNA (Промпт как ДНК)

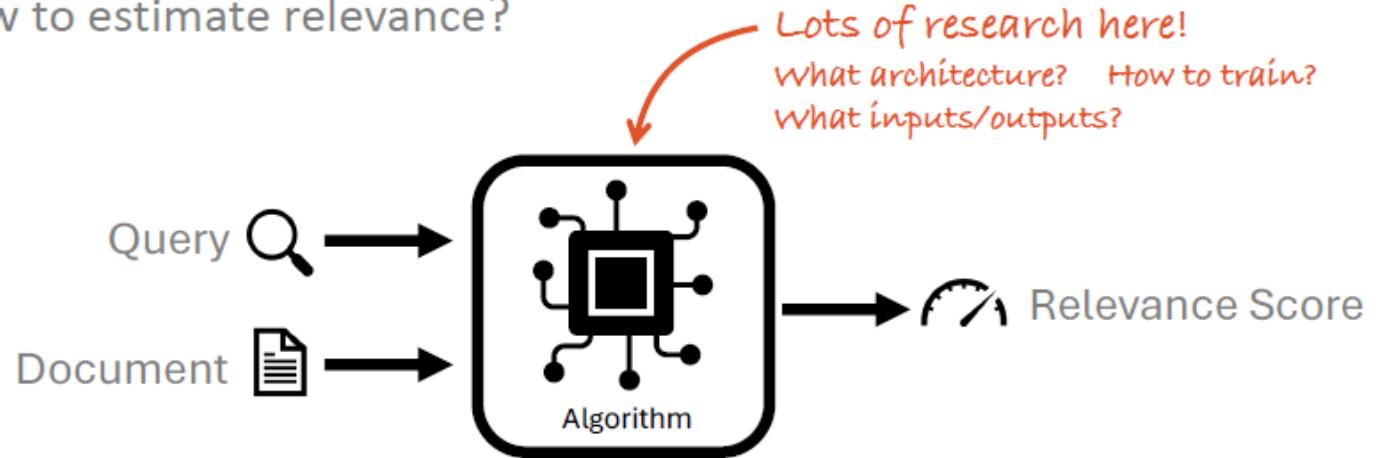
вопросы



RAG (Long-term memory)

Relevance Models

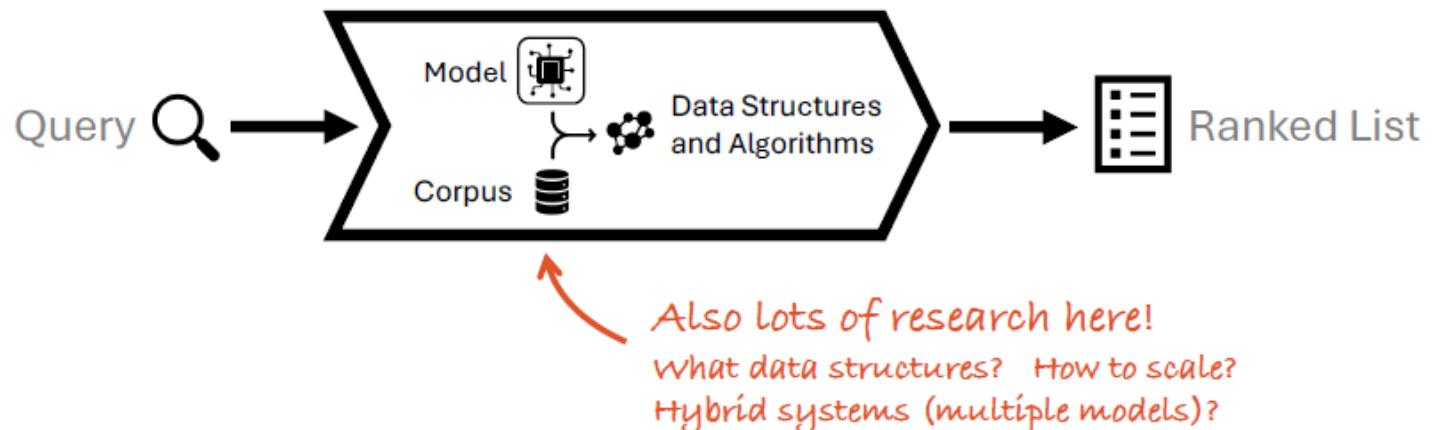
How to estimate relevance?



Retrieval Engines

How to find the top results for a relevance model?

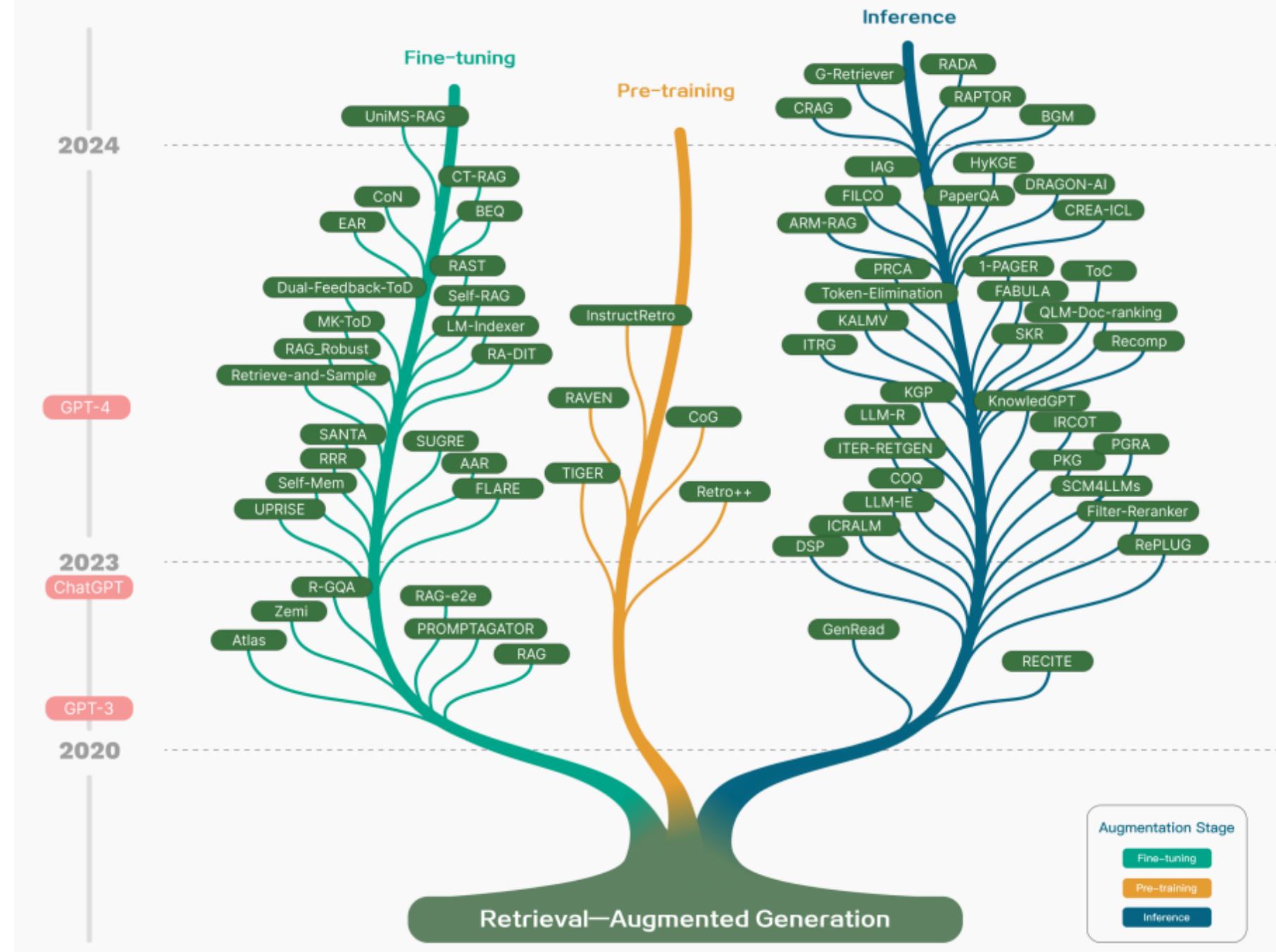
https://drive.google.com/file/d/1R5y0iPdm9zpbVIWoVDctPTPZBucoZ_jT/view?pli=1

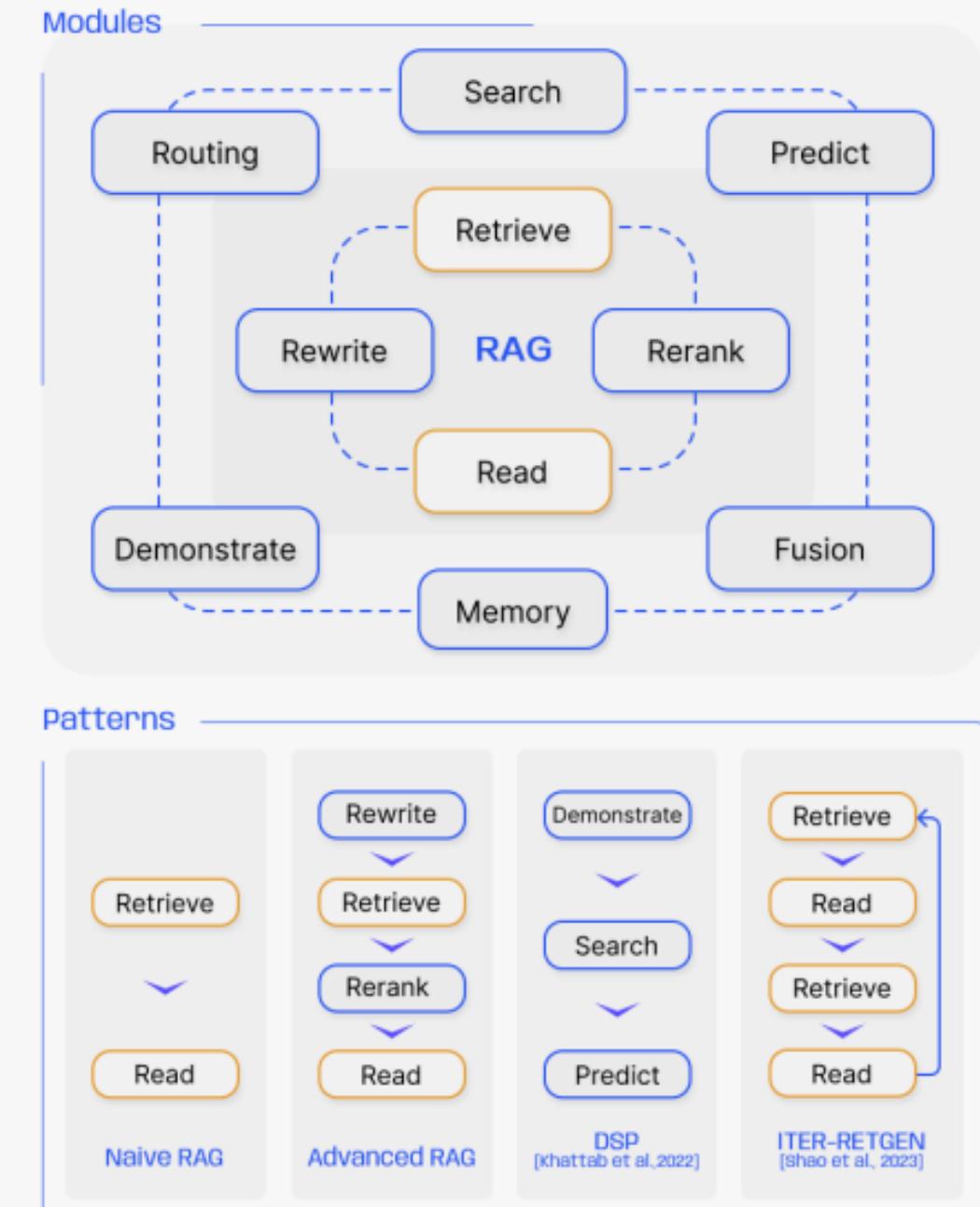
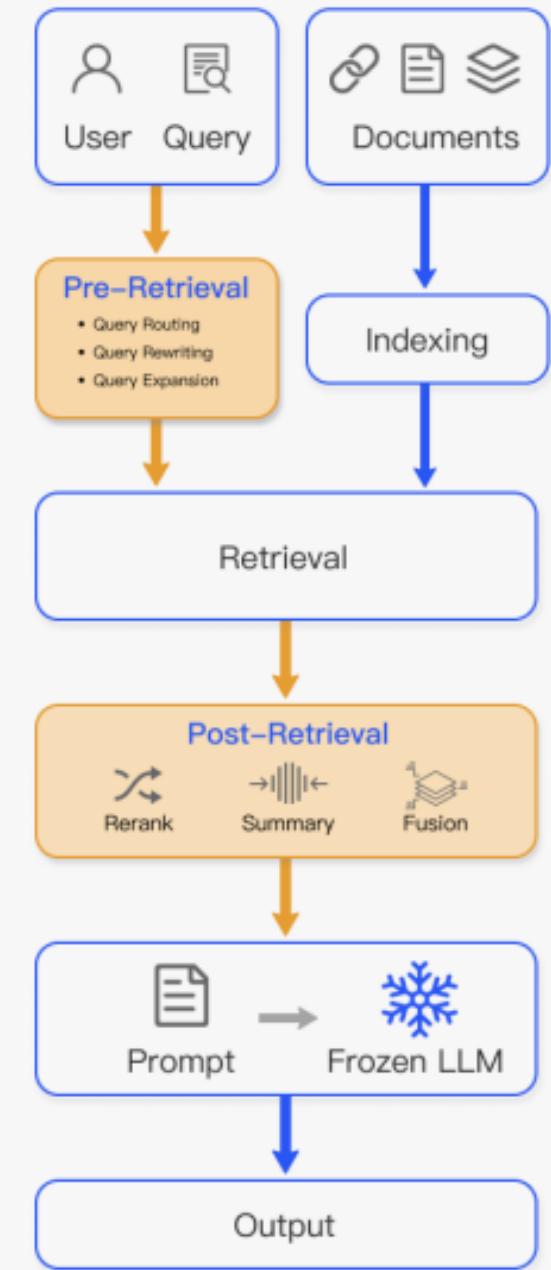
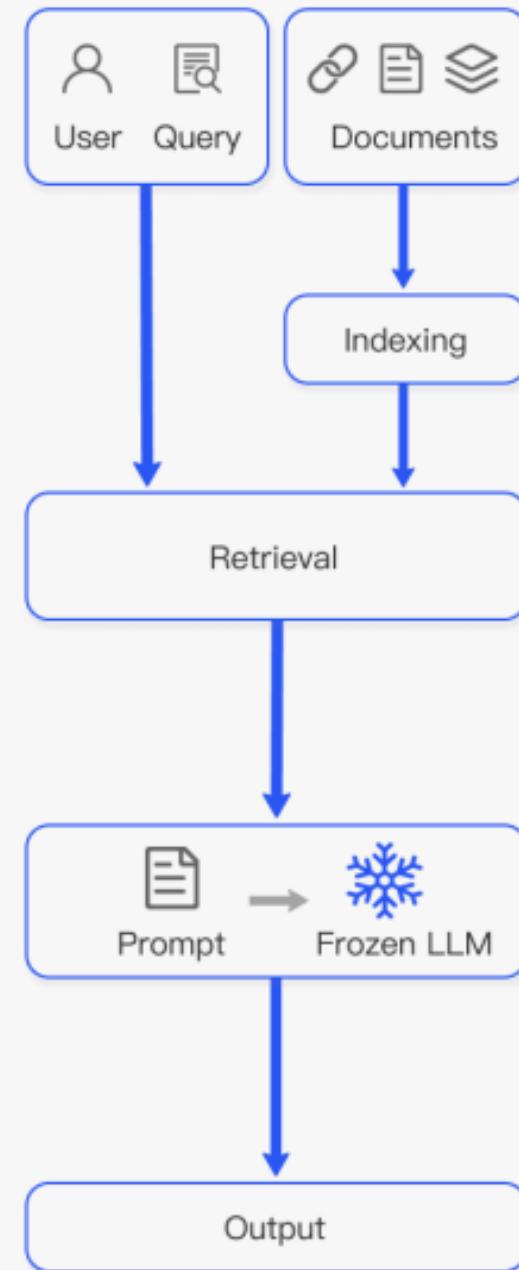


RAG (Long-term memory)

A Survey

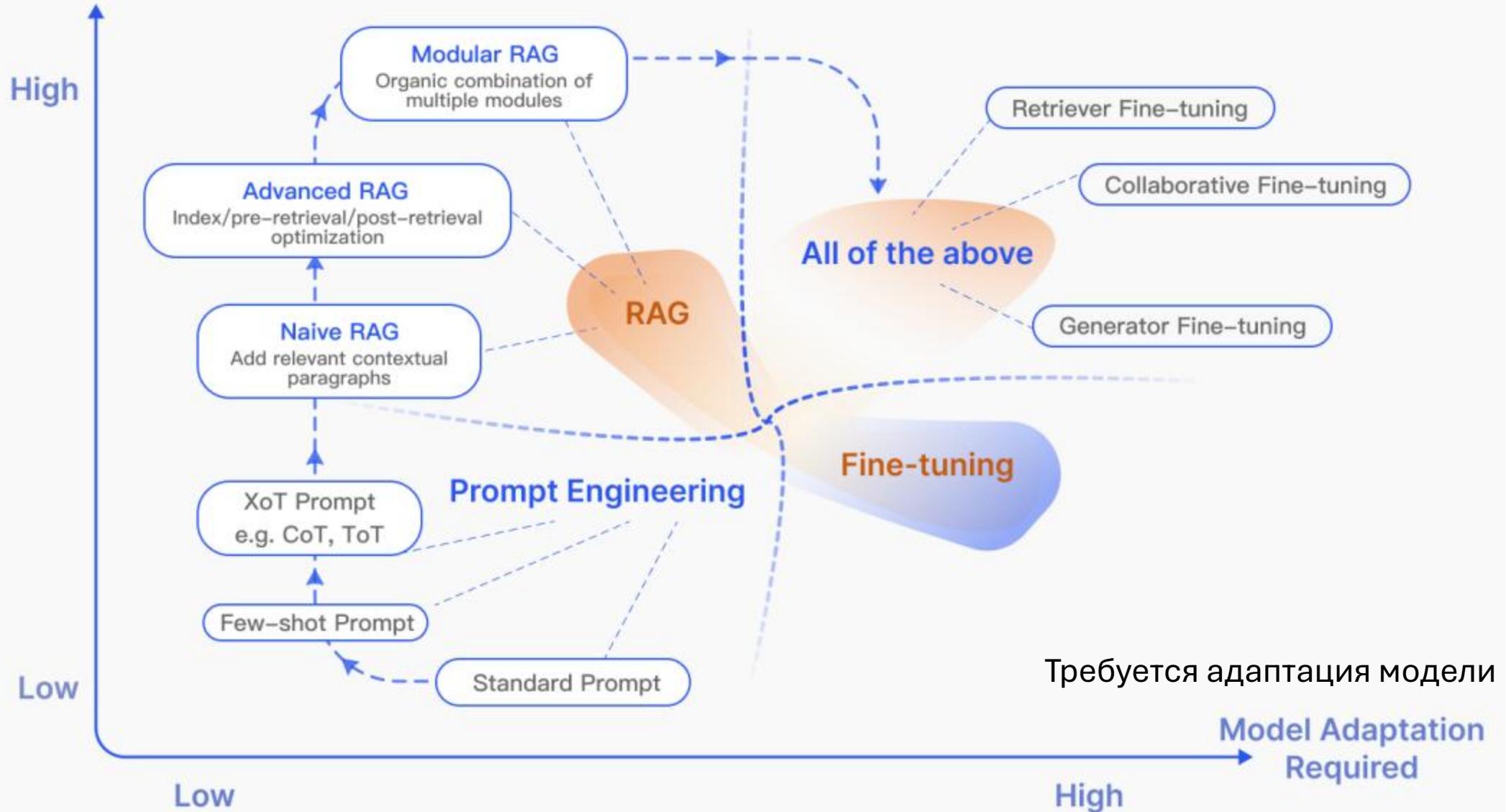
<https://arxiv.org/pdf/2312.10997.pdf>





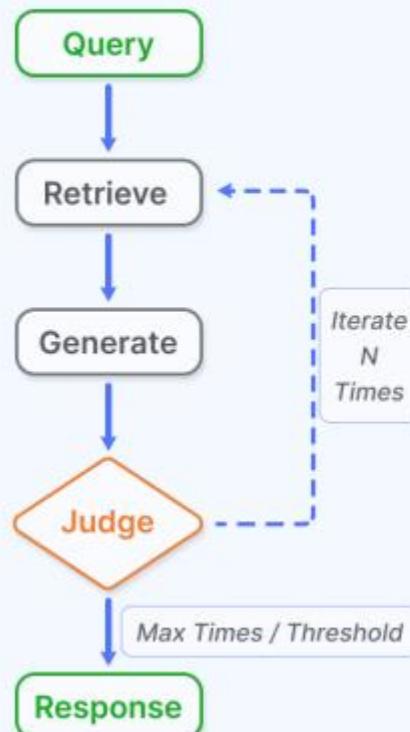
Требуется внешнее знание

External Knowledge Required



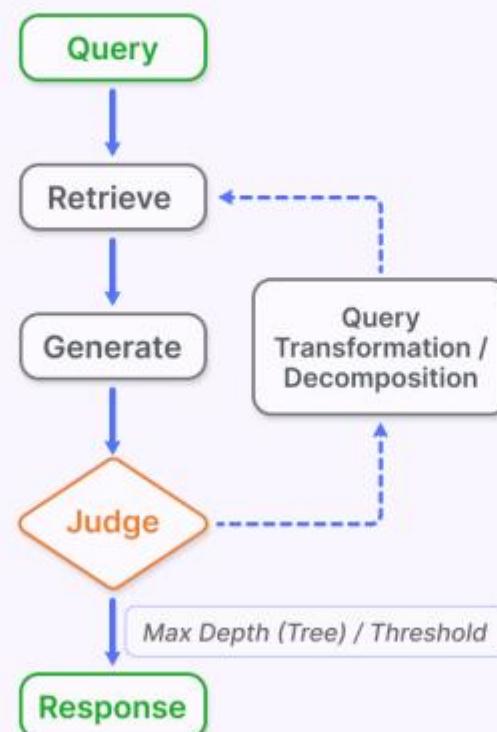
ITERATIVE

Provide more context information



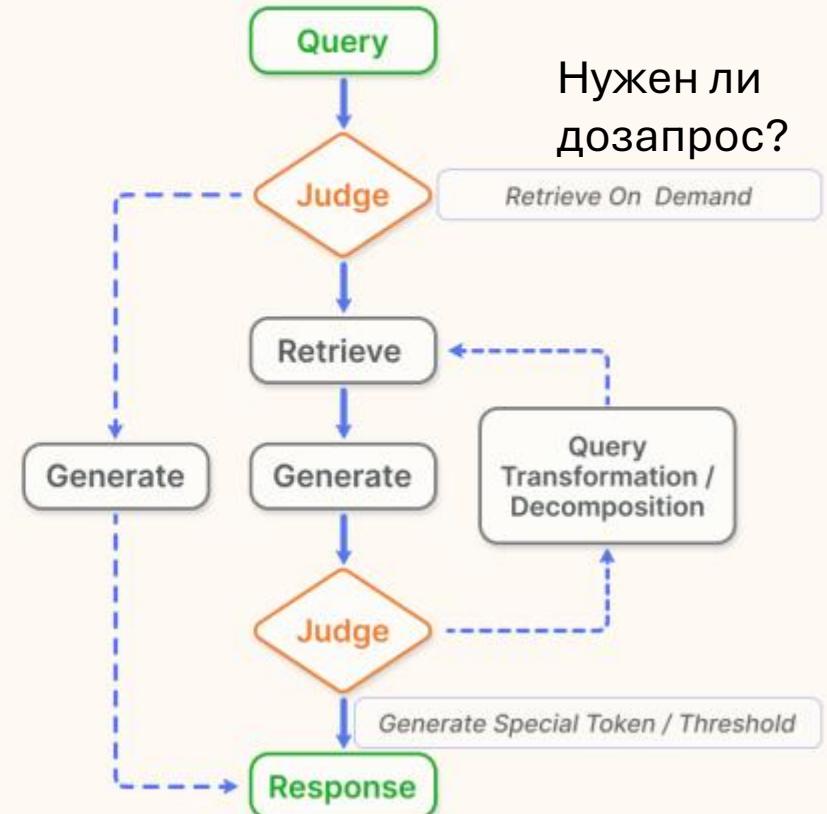
RECURSIVE

Break down complex problems step by step



ADAPTIVE

Flexible and active control of retrieval and generation

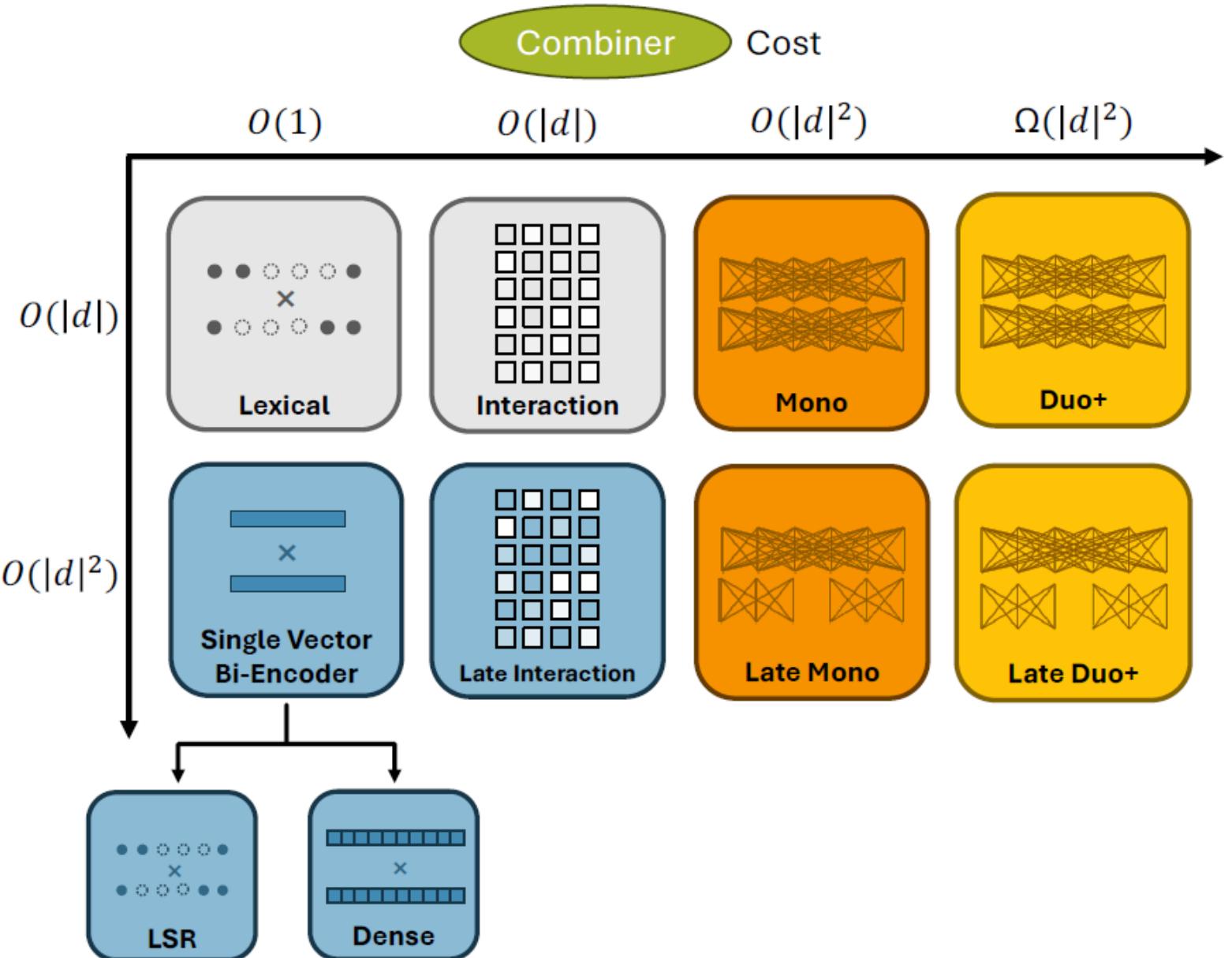
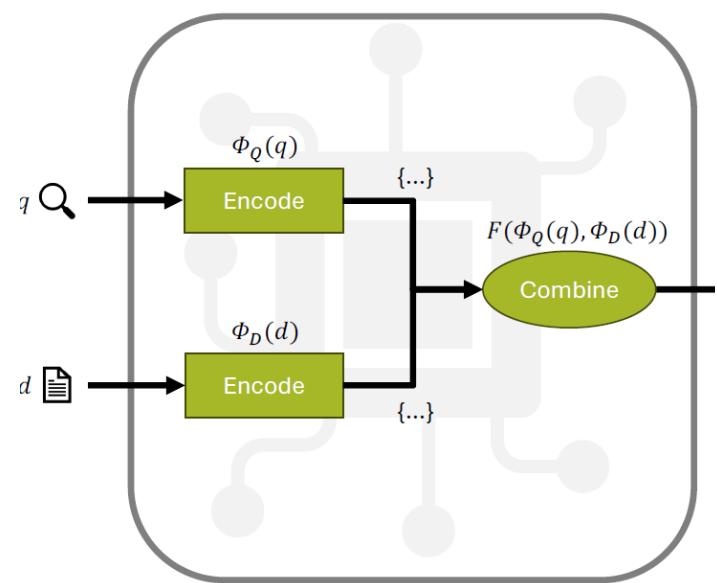


вопросы



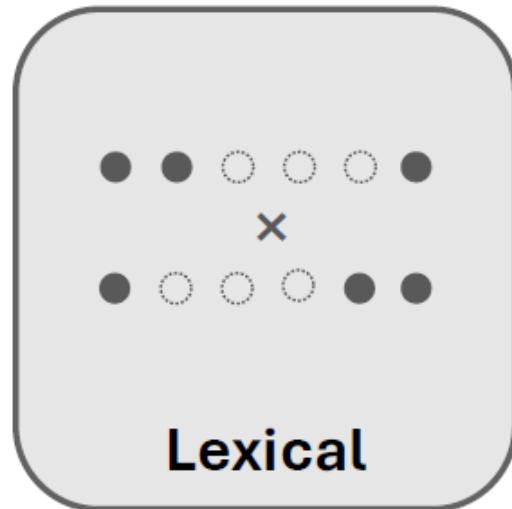
The Table of Relevance Models

Особенности эмбедов



Lexical Relevance Models

Особенности эмбедов



Проблема: важность слов и контекст не учитываются, не учитываются синонимы,
...
...

Representations based on lexical form

Combination based on overlapping terms

Variations:

- Stopword Removal
- Stemming
- Inverse Document Frequency (IDF)
- Sublinear Term Frequencies
- Document length normalization
- ... Others too!

Common Lexical Relevance Models:

- TF-IDF Cosine
- BM25
- Query Likelihood (QL)
- Divergence from Random
- ... Others too!

Особенности эмбедов

+ учет важности терминов (за счет механизма внимания).

Learned Sparse Retrieval (LSR)

Representations learned and aligned to lexicon

(Выделяем «важные» токены среди всего текста)

Combination based on overlapping terms
(potentially expanded)

Variations:

Token weighting

Expansion (per-token, single-token, external)

Sparsification (top-k, regularization)

... Others too!

Common LSR models:

DeepCT [DeepCT]

EPIC [EPIC]

SPLADE [SPLADE]

TILDE [TILDE]

... Others too!

DENSE

Representations learned as fixed-size dense vectors (usually

~500-2000 dimensions) (Берем эмбеды от слов, mean
pooling + cos sim)

Combination as similarity or distance function between
vectors

Variations:

Pooling: CLS (common) vs mean token vector

Combination: dot product (common) vs cosine vs Euclidean

Symmetric vs Asymmetric Encoders

Common Single-Vector Dense models:

DPR [DPR]

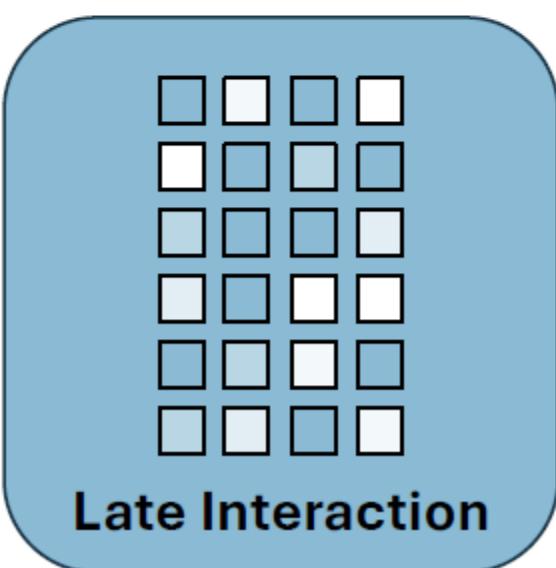
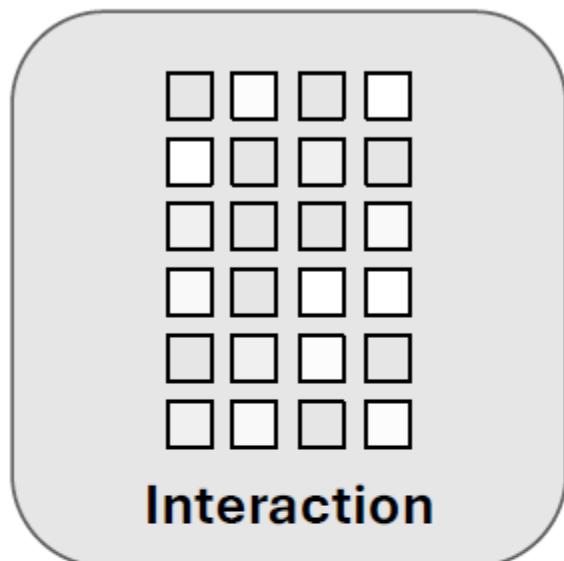
TCT-CoBERT [TCT]

(Lots more, but they differ in training procedures, not
architectures)

ANN – для поиска => O(1)

(Late) Interaction Models

Point-wise



Lightweight “interaction” combination over per-token query/doc vectors

Encoders either static or contextualized (“late” interaction)

Variations:

Combination function: MaxSim (common) vs kernel pooling vs ...

Encoding strategy: LM (Late Interaction), Static (Interaction)

Common Interaction and Late Interaction models:

ColBERT [ColBERT]

Близость рассчитывается для каждой пары эмбедов от слов из запроса и документа, потом берется макс для токенов из запроса и сумма.

DRMM [DRMM]

Late: После эмбедов есть еще слои (подробнее потом).

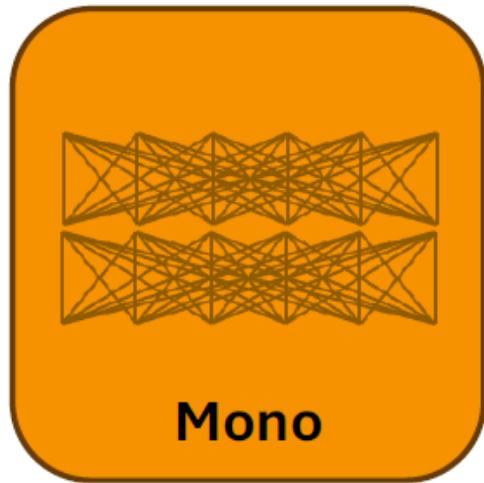
KNRM [KNRM]

PACRR [PACRR]

CEDR [CEDR]

Others...

Pair-wise



Uses cross attention between query and document for combination

Minimal separate representations (i.e. just tokenization)

Variations:

Architecture: Encoder Only, Encoder-Decoder, Decoder Only

Attention Pruning: Skip some relations [SparseAttn]

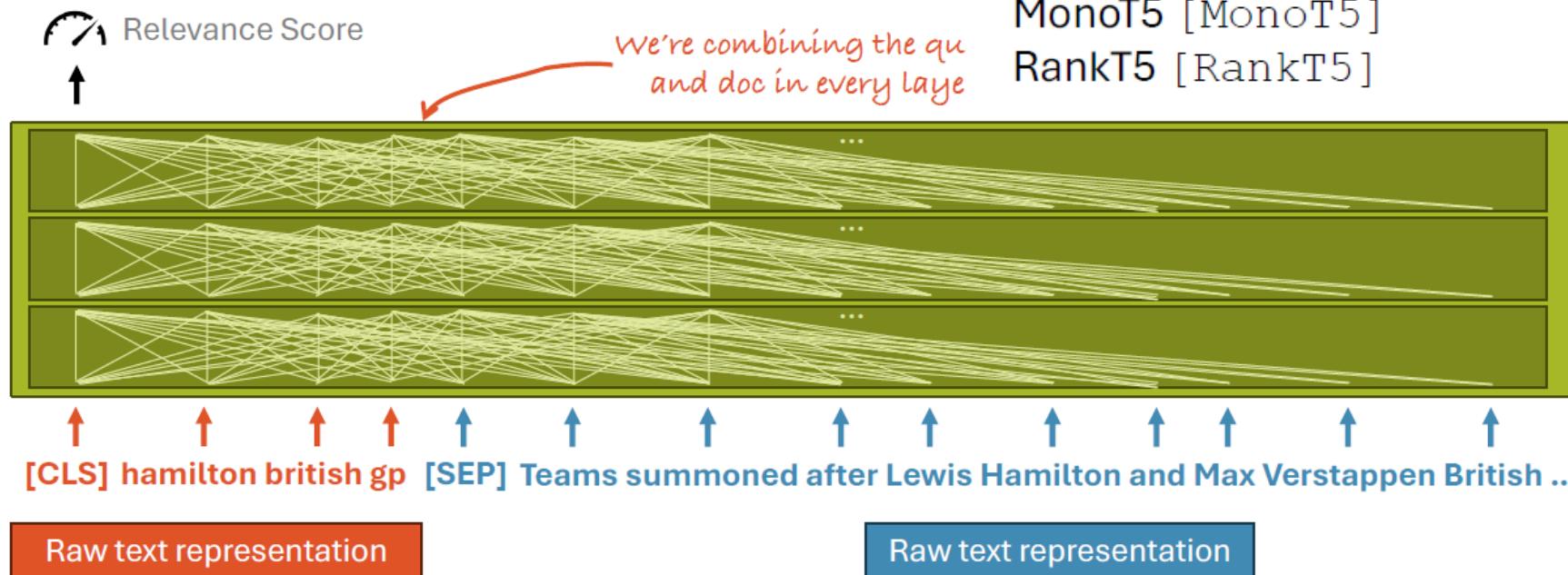
Model Size: Shallow networks [ShallowCE]

Common models:

MonoBERT / BERT-cat / Vanilla BERT [BertRR]

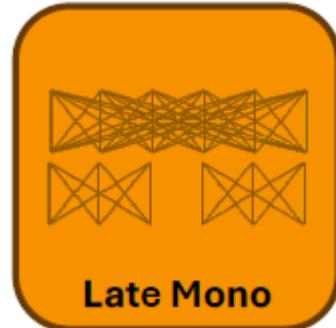
MonoT5 [MonoT5]

RankT5 [RankT5]



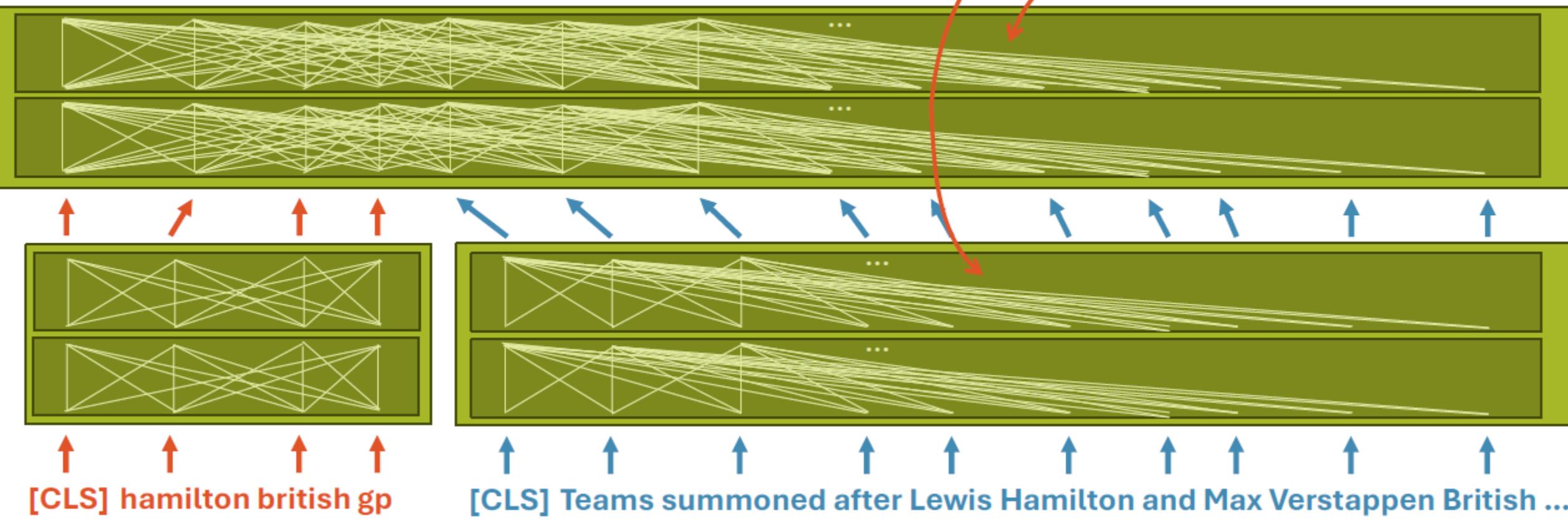
Pair-wise

Relevance Score
↑



Пример: PreTTR

Some separate encoding,
some combined



Uses cross attention between query and **multiple** documents for combination

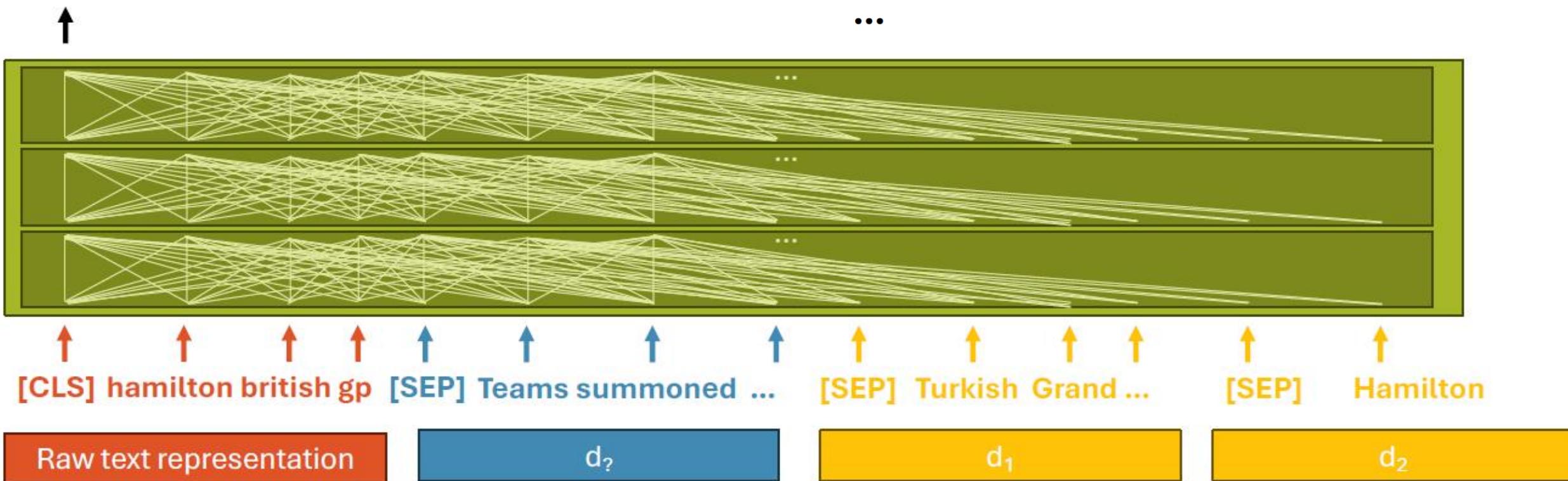
More on these in the LLMs for ranking sessions!

Common models:

- DuoBERT [DuoBERT]
- ListoBERT [ListoBERT]
- RankGPT [RankGPT]
- PE-Rank [PERank]
- ... More!

List-wise

Relevance Score



Опции ml-решения с RAG

- Как нарезать док на чанки (разделитель, семантика, люди,...)?
- Иерархическое разбиение документов (документ -> разделы -> подразделы)
- Мета-данные о чанке (например, от html)
- Переписать чанк
- Связь чанков между собой (графы?)
- Количество чанков
- Размер чанков
- Перекрытие чанков
- Добавление онтологии (релевантных терминов), уникального контекста о пользователе
- Реранжирование чанков по новизне, diversity, ...
- Каскадная система поиска (лексика -> bi-encoder -> cross-encoder)
- Обновлять базу данных не полностью, а лишь инкрементально; Предиктивное обновление
- Отдельная обработка специальных типов контента (таблиц, структур, ...)
- Использовать прямые цитаты для фактической привязки: сначала просить извлечь дословные цитаты, прежде чем выполнять задачу. Это "привязывает" ответ к тексту.
- Проверять с помощью цитирования:
 - Делать выводы LLM проверяемыми, требуя указывать цитаты и источники для каждого утверждения.
 - Альтернативный метод: Просить LLM после генерации ответа проверить каждое утверждение, найдя подтверждающую цитату. Если цитата не найдена - утверждение должно быть отозвано.
- Thread of Thought (ThoT) – целый план по анализу контекста
- Chain of Note (CoN) – делать последовательные заметки с оценкой достоверности
- Roaming RAG – не нарезать на чанки, а дать LLM структуру документа для самостоятельного исследования



КОНЕЦ ЛЕКЦИИ 28.02