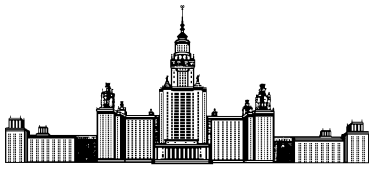

Московский государственный университет имени М.В. Ломоносова Факультет
Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Пукемо Михаил Михайлович

Выявление галлюцинаций и связанных с ними наблюдаемых ошибок сверхгенерации на английском языке у больших языковых моделей

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., профессор
Воронцов К.В.

Москва, 2025

Содержание

Аннотация

В данной работе рассматривается проблема галлюцинаций в больших языковых моделях (LLM) и предлагаются методы для их обнаружения и минимизации. Галлюцинации, или генерация недостоверной, вводящей в заблуждение информации, остаются критической проблемой для LLM, особенно в контексте приложений, требующих высокой точности, таких как медицинская и юридическая документация. В работе проанализированы различные подходы, включая Retrieval-Augmented Generation (RAG), Chain of Thought (CoT), самоуточнение (self-refine) и ансамблирование моделей.

Эксперименты, проведенные на наборе данных SemEval-2025 (Задача 3 - Mu-SHROOM), показали, что комбинация методов RAG, CoT и ансамблирования позволяет значительно повысить точность обнаружения галлюцинаций. В частности, использование ансамблирования с CoT и RAG достигло лучших результатов, превысив 60% по метрикам IoU и корреляции Пирсона. Самоуточнение, напротив, не всегда улучшало результаты, что может быть связано с особенностями архитектуры моделей.

Результаты работы подчеркивают важность интеграции внешних источников знаний и многоэтапных методов рассуждения для повышения надежности LLM. Предложенные подходы могут быть полезны для разработки более устойчивых языковых моделей, что особенно актуально для приложений, где точность информации является критической.

1 Введение

Большие языковые модели (LLM) произвели революцию в обработке естественного языка (NLP), продемонстрировав впечатляющие возможности в генерации текста, резюмировании и диалоговых системах. Однако LLM всё ещё подвержены искажениям, когда сгенерированный контент содержит ложную, вводящую в заблуждение или непроверяемую информацию, которую называют "галлюцинацией" (10), (25).

Детекция галлюцинаций является критически важной задачей для обеспечения надёжности и точности языковых моделей, особенно в приложениях (8), где точность информации является ключевой, таких как медицинская (13) или юридическая документация (4). Потенциальные риски, связанные с использованием моделей, которые могут генерировать недостоверную информацию, подчеркивают необходимость разработки методов для их выявления и устранения.

Галлюцинации могут происходить по разным причинам (15), таким как недостаточные обучающие данные, ошибки в архитектуре модели или вводные запросы, содержащие вводящую в заблуждение или неоднозначную информацию. Обнаружение галлюцинаций является важной задачей в обработке естественного языка (NLP) и стало важным критерием оценки устойчивости языковых моделей (29).

Одним из самых популярных тестов для обнаружения галлюцинаций является HADES (17). HADES предназначен для классификации на уровне токенов, где метки представляют собой фрагменты галлюцинирующего текста. Набор данных состоит из фрагментов текста из статей Википедии, которые были помечены людьми как галлюцинирующие или негаллюцинирующие. Наиболее распространённый тип галлюцинаций, выявленный в наборе данных, — это использование собственных имён, которые легко спутать с реальными сущностями (Рис **??**). Исследователи обучили модели, которые могут обнаруживать галлюцинирующие фрагменты и помогают понять характеристики галлюцинаций в тексте.

Однако галлюцинации могут возникать, несмотря на знания и обучающие данные модели (26). Это может быть связано с плохими вводными запросами, которые могут содержать много нерелевантной или вводящей в заблуждение информации. Например, если модель просят обобщить сложную научную статью, она может галлюцинировать, выдумывая детали, не подтверждённые оригинальным текстом (1). Для решения этой проблемы исследователи разработали методы создания наборов данных, которые специально фокусируются на галлюцинациях, вызванных плохими вводными запросами

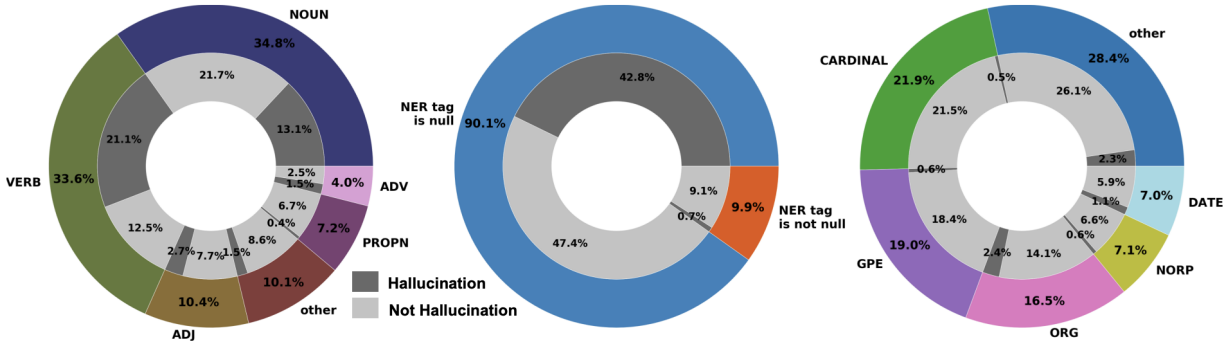

Рис. 1: Распределение галлюцинаций по частям речи

(2). Эти наборы данных помогают анализировать ошибки моделей и разрабатывать процедуры обучения, которые делают LLM более устойчивыми к входному шуму (1).

Обнаружение галлюцинаций также стало популярной задачей для конкурсов и оценочных соревнований. SemEval 2024, ведущая конференция по NLP, включала задачу по обнаружению галлюцинаций (21). Участникам было предложено классифицировать, содержат ли выходные данные модели галлюцинации или нет. Лучшие решения (6) этой задачи включали использование GPT для генерации меток для большого набора данных и последующее обучение меньших моделей, техника, известная как слабо контролируемая тонкая настройка (28). Интересно, что ансамбли меньших моделей, обученные на разметке GPT, часто превосходили саму GPT (27), демонстрируя эффективность этого подхода в обнаружении галлюцинаций.

Задача SemEval-2025, Задача 3 — Mu-SHROOM[1](30)редставляет собой многоязычный вызов по обнаружению искажений. Участникам необходимо определить конкретные фрагменты искажённого текста в выходных данных модели. В отличие от традиционных задач по проверке фактов, Mu-SHROOM предоставляет текст, созданный LLM, вместе с токенизированными представлениями и логарифмическими оценками. Участники должны вычислить вероятность для каждого символа, указывающую на вероятность его принадлежности к искажению.

Задача охватывает 14 языков, включая английский, китайский, арабский и несколько европейских языков. Это представляет уникальные вызовы, такие как языковое разнообразие, межъязыковые закономерности искажений и вариации в поведении моделей.

Извлечение с помощью дополненной генерации (Retrieval Augmented Generation, RAG) (5), (24) — это самый популярный метод для уменьшения галлюцинаций. Основ-

[1]https://helsinki-nlp.github.io/shroom/

ная идея этого подхода заключается в организации внешних данных в базе данных и преобразовании пользовательского ввода в запрос для получения полезной информации. Информация из базы данных добавляется в качестве контекста к запросу для больших языковых моделей. Система, которая извлекает информацию из базы данных, называется системой извлечения. Самая сложная часть этого метода — сделать извлечение быстрым и обеспечить качественный поиск по документам. Самые популярные методы представлены на изображении (Рис 2).

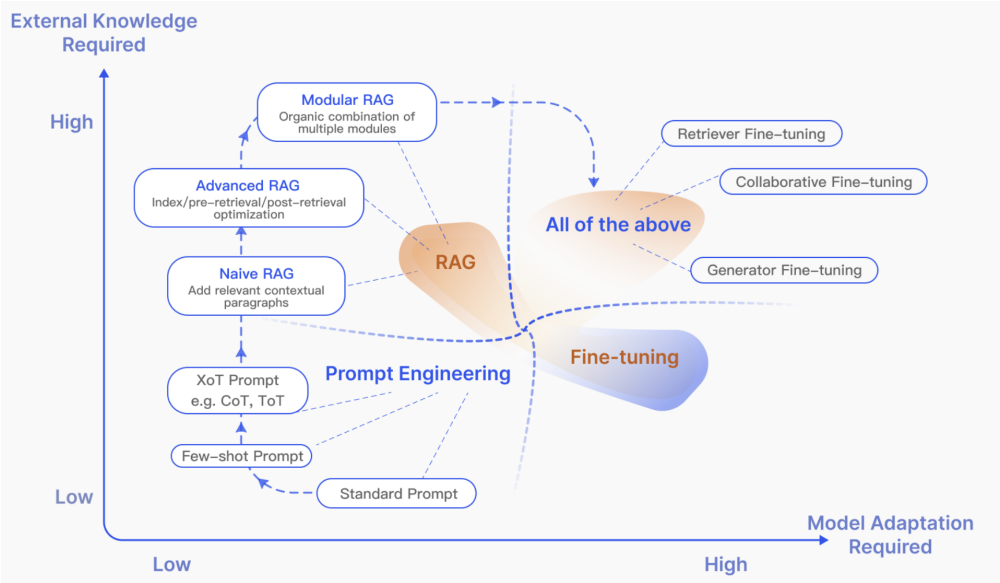

Рис. 2: Сравнение RAG с другими методами оптимизации моделей по аспектам "Требуемые внешние знания" и "Требуемая адаптация модели". Промпт-инжиниринг требует минимальных изменений модели и внешних знаний, делая акцент на использовании возможностей самих больших языковых моделей (LLM). Тонкая настройка (fine-tuning), напротив, предполагает дополнительное обучение модели. На ранних этапах RAG (наивный RAG) потребность в модификациях модели невелика. По мере развития исследований модульный RAG становится всё более интегрирован с техниками тонкой настройки.

Помимо необходимой для выявления галлюционаций информации нужна модель для обнаружения галлюцинаций. Во многих исследованиях используют дообучение языковых моделей на основе кодировщиков, таких как BERT(12), roBERTa(19), deBERTa(9). Для тонкой настройки кодировщиков требуется большая обучающая выборка.

С ростом популярности генеративных моделей таких как GPT(18), DeepSeek(16), широкое распространение получили подходы с написанием инструкций, так называемых промптов.

Один из базовых методов написания инструкций - Few-shot prompting(14). Это мето-

дика, связанная с использованием языковых моделей, особенность которой заключается в способности модели выполнять задачи с минимальным количеством примеров. В контексте обработки естественного языка, это позволяет моделям эффективно работать с новыми или редкими задачами (22), получая всего несколько иллюстративных примеров для достижения приемлемого уровня производительности.

Методика few-shot prompting приобретает огромное значение в современном мире, где задачи могут быстро изменяться, а данные не всегда доступны в обильных количествах (3). Возможность модели учиться на ограниченном числе примеров делает её не только экономически более привлекательной, но и более применимой в практических сценариях. Например, в сфере здравоохранения (23), где данные пациентов могут быть ограничены и конфиденциальны, способность модели адаптироваться и работать эффективно с минимальной информацией чрезвычайно важна.

Среди подходов выделяется цепь размышлений (chain of thought reasoning)(33).

Chain of thought reasoning позволяет улучшить качество генерируемых ответов за счёт поэтапного разложения задачи на последовательность промежуточных шагов. Эта техника становится особенно актуальной в контексте больших языковых моделей, которые часто требуют более глубокого уровня понимания и интерпретации для решения сложных задач.

Внедрение chain of thought reasoning в большие языковые модели требует соответствующей архитектуры и обучения, чтобы модель могла эффективно разлагать задачи и использовать такие разложения в процессе генерации. Исследователи продолжают изучать способы интеграции этого подхода для оптимизации производительности и точности моделей в различных приложениях, включая автоматическую проверку фактов, решение математических задач и многими другими сферами, где глубокое понимание и логическое рассуждение играют ключевую роль.

Самые популярные архитектуры: o1(11) от OpenAI и R1(7) от DeepSeek будут рассмотрены в данной работе.

Развитие данной техники - Self-Consistency with Chain of Thought (CoT-SC) представляет собой передовой метод декодирования, значительно улучшающий возможности языковых моделей в решении задач, требующих сложных рассуждений. Данный подход, предложенный в исследовании (32), позволяет преодолеть ограничения базового метода цепочки размышлений (Chain of Thought), существенно повышая точность ответов моделей на математические задачи, задачи логического вывода и здравого смысла. Ключевая идея метода заключается в генерации множественных путей рассуждения и выборе

наиболее согласованного ответа среди них, что приводит к впечатляющим улучшениям производительности – до 17,9% на задачах математического рассуждения и до 6,4% в задачах здравого смысла.

Метод Self-Consistency основывается на важном наблюдении: при решении сложных задач существует множество различных правильных путей рассуждения, которые приводят к одному и тому же верному ответу. В то же время, неправильные пути рассуждения с меньшей вероятностью сойдутся к одинаковому результату. Традиционный CoT-подход использует жадное декодирование (greedy decoding), которое генерирует только один путь рассуждения, что ограничивает потенциал модели.

Self-Consistency преодолевает это ограничение, заменяя жадное декодирование более сложной стратегией. Вместо генерации единственной цепочки рассуждений, метод предполагает отбор разнообразных путей решения задачи с последующим выбором наиболее согласованного ответа. По сути, CoT-SC "маргинализует"различные пути рассуждений, что позволяет модели использовать преимущества множественных подходов к решению проблемы.

Метод Tree of Thoughts (ToT) (34) представляет собой прорыв в области промпт-инжиниринга, значительно расширяющий возможности языковых моделей (LLM) в решении многокомпонентных задач, требующих стратегического планирования и комплексного анализа. Разработанный в 2023 году, этот подход преодолевает ограничения традиционных линейных методов, таких как Chain of Thought (CoT), за счёт внедрения древовидной структуры рассуждений. ToT демонстрирует впечатляющие результаты: в задаче Game of 24 успешность решения возрастает с 4% при использовании CoT до 74% с применением ToT, что подтверждает его эффективность в задачах, требующих нестандартного мышления.

Основное отличие ToT от Chain of Thought заключается в переходе от линейного последовательного рассуждения к ветвящейся структуре возможных решений. Если CoT следует единственному пути рассуждений, подобно цепочке, то ToT создаёт "дерево"альтернативных вариантов, каждый из которых представляет собой потенциальное промежуточное решение.

Разновидности представлены на изображении (Рис. 3).

Другая техника для улучшения - самоуточнение (self-refine). Техника самоулучшения ответа (20) представляет собой подход в области искусственного интеллекта, направленный на повышение качества и точности ответов, генерируемых моделями, с помощью итеративного процесса саморецензирования и корректировки. Основная идея заключа-

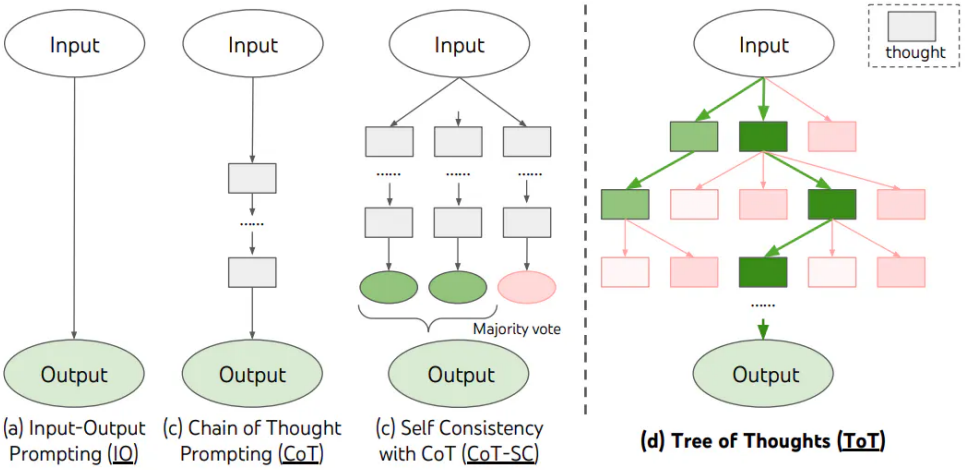

Рис. 3: Разновидности CoT

ется в том, чтобы модель сама проверяла свои ответы и вносила улучшения на основе обнаруженных недостатков или ошибок.

В рамках данной работы рассматриваем решение на основе дополненной генерации, цепи размышлений. Качество решения продемонстрировано на соревновании SemEval-2025, Задача 3 — Mu-SHROOM.

2 Постановка задачи

2.1 Абстрактная постановка

Задача поиска фрагментов с галлюцинациями в больших языковых моделях связана с выявлением и исправлением тех частей генерируемого текста, которые содержат недостоверную, вымышленную или искаженную информацию. Галлюцинации возникают, когда модель создает контент, который выглядит правдоподобно и уверенно, но по сути не соответствует фактической реальности. Этот феномен представляет серьёзную проблему, особенно в приложениях, где точность и достоверность данных критически важны.

Необходимо определить, какие именно фрагменты текста считаются галлюцинациями. Обычно это выражения, содержащие неточные факты, ложные утверждения или вымышленные сведения, которые модель выдает с высокой степенью уверенности.

2.2 Математическая постановка

Пусть заданы:

- Запрос пользователя (query, промпт, задача) — последовательность токенов ($X = \{x_1, \ldots, x_l\}$), где l — длина последовательности.
- Ответ модели — последовательность токенов ($Y = \{y_1, y_2, \ldots, y_n\}$), где n — длина последовательности.

Задача детекции галлюцинаций заключается в поиске всех пар (i_k, j_k), где i_k и j_k такие, что $1 \leq i_k \leq j_k \leq n$, представляющих собой индексы начала и конца фрагмента $\overline{y_{i_k} y_{i_k+1} \ldots y_{j_k}}$, являющегося галлюцинацией.

3 Методика

Основа решения - использование генеративных языковых моделей. Для эффективного использования генеративных моделей нужно использовать техникик написания инструкций (промптов).

3.1 Методики промптизации моделей

3.1.1 только LLM

В данном подходе мы оцениваем обнаружение галлюцинаций с использованием автономных LLM без дополнительного поиска только на валидационном наборе данных. В частности, мы экспериментируем с Qwen2.5-72B, GPT-4o, используя подсказки.

В промпт добавляется инструкция где объясняется суть задачи и в каком виде возвращать результат в виде json-конструкции.

3.1.2 Few-shot

Few-shot — это термин в контексте машинного обучения и особенно в области обучения моделей на основе трансформеров, таких как большие языковые модели (LLM), который описывает способность модели эффективно выполнять задачу, предоставив ей лишь несколько примеров для обучения.

Few-shot обучение подразумевает, что модель может адаптироваться к новой задаче, используя очень ограниченное количество обучающих примеров (например, один, два или несколько десятков), что делает данный подход крайне эффективным в соревновании SemEval, собрано очень мало данных.

3.1.3 One-shot

One-shot — Частный случай few-shot с одним примером. Хотя few-shot подход (использование нескольких примеров в промпте) традиционно повышает качество генерации языковых моделей, в случае RAG-систем (Retrieval-Augmented Generation) переход от one-shot к few-shot зачастую не оправдывает себя. Во-первых, увеличение числа примеров быстро съедает лимит длины контекстного окна, что особенно критично при работе с длинными retrieved-документами; важная retrieved-информация может попросту не попасть в prompt. Во-вторых, согласно ряду исследований, при RAG архитектуре добавление дополнительных примеров (промтов) даёт лишь незначительный прирост

качества по сравнению с one-shot—модель и так располагает достаточным количеством корректных знаний из retrieval-компоненты. Таким образом, one-shot оптимален с точки зрения баланса между эффективным использованием контекста и качеством вывода для RAG.

3.1.4 Конвейер RAG

Для повышения точности обнаружения галлюцинаций мы реализуем конвейер RAG, используя Qwen2.5-72B в качестве LLM и векторную базу данных, как подробно описано в разделе "Набор данных и база данных для RAG".

С помощью языковой модели E5 для каждого документа получаем векторное представление (эмбеддинг). По вектору запроса находим релевантные вики-страницы и дополняем ими промпт.

3.1.5 Самоуточнение

Для дальнейшего улучшения обнаружения галлюцинаций мы применяем стратегию самоуточнения (Self-Refine Prompting), где модель оценивает и уточняет свои собственные генерации. Мы используем Qwen2.5-72B в качестве LLM для уточнения вывода, созданного нашим конвейером RAG.

Данный приём можно использовать многократно, но в данной работе используется только однократное применение.

3.1.6 Цепь размышлений

Другая техника - цепь размышлений (Chain of Thought reasoning). Chain of thought reasoning — это методика, используемая в больших языковых моделях для улучшения их способности решать сложные задачи путем поэтапного представления хода рассуждений. Основная идея заключается в том, чтобы вместо того, чтобы напрямую давать ответ на вопрос, модель сперва генерировала промежуточные шаги рассуждений, которые помогают прийти к обоснованному решению. Это позволяет моделям более эффективно справляться с задачами, требующими логического или математического мышления.

Основные аспекты chain of thought reasoning:

- Поэтапное мышление: В данной технике моделям предлагается разбивать сложные задачи на несколько более простых шагов. Например, при решении арифметической

задачи модель может сначала вспомнить формулы, применить их к конкретным параметрам задачи и только потом прийти к окончательному результату.

- Транспарентность: Благодаря явному перечислению промежуточных шагов пользователи могут видеть, как модель пришла к своему выводу, что делает процесс принятия решения более прозрачным и понятным.

- Улучшение точности: Поскольку модель обязана проходить через несколько этапов размышлений, это помогает избежать распространённых ошибок, которые могут возникнуть при попытке дать единичный, прямой ответ без анализа.

- Применимость в различных областях: Chain of thought reasoning полезна в различных приложениях, например, в задаче решения математических заданий, логических вопросов, головоломок и других областях, требующих строгого следования шагам рассуждений.

Пример:

Рассмотрим задачу: «У Пети было три яблока, он купил ещё пять. Сколько всего яблок у Пети сейчас?»

Вместо прямого ответа модифицированная на chain of thought reasoning модель может сначала вывести:

- Петя начал с трех яблок.

- Он купил ещё пять яблок.

- Таким образом, у него теперь три плюс пять яблок.

- В конечном итоге это восемь яблок.

Таким образом, техника chain of thought reasoning помогает моделям обеспечивать более обоснованные и точные ответы, избегая поверхностного подхода к решению задач. Это особенно актуально для задач, требующих многоэтапных действий или глубокой аналитики.

Основная идея состоит в том, чтобы стимулировать модель к созданию межзвенных логических или причинно-следственных связей при решении задач, особенно тех, которые требуют многократного рассуждения. В данной работе рассматриваем как CoT помогает лучше детектировать фрагменты.

3.1.7 Техника ансамблирования

Мы получаем набор жёстких меток, которые представляют собой список индексов, соответствующих галлюцинированным фрагментам.

Чтобы объединить несколько выходов, мы преобразуем эти жёсткие метки в мягкие, которые представляют собой вероятности галлюцинаций для каждого символа.

Для вычисления вероятности галлюцинации для каждого символа мы определяем долю моделей, которые отметили его как часть галлюцинированного фрагмента.

Финальный ответ представляется с использованием кодирования длины диапазона этих вероятностей. Дополнительно мы удаляем галлюцинации из одного символа и все знаки препинания из галлюцинаций.

Пример (Рис. 4): Дано предложение, сгенерированное моделью:

"Петра ван Стоверен завоевала серебряную медаль на летних Олимпийских играх 2008 года в Пекине, Китай."

Три модели возвращают разные фрагменты галлюцинаций:

- Модель 1: "серебряная"
- Модель 2: "серебряная медаль"
- Модель 3: "завоевала серебряную медаль"

Финальное предсказание ансамбля присваивает вероятности: "завоевала"→ 0.33 "серебряная"→ 1.00 "медаль"→ 0.66

Для системы оценки этот вывод записывается как список индексов, соответствующих вероятностям галлюцинаций.

Данная техника позволяет агрегировать ответы моделей и повышать качество выделения галлюцинаций. Чем разнообразнее модели, промпты, тем лучше финальный ответ. Поэтому ансамблирование происходит после экспериментов с остальными промптами.

3.2 Финальное решение

На изображении (Рис. 6) представлено финальное решение, которое помогло добиться 1 места в соревновании SemEval2025-task3. Оно включает в себя техники: RAG, CoT, Refine и ансамблирование.

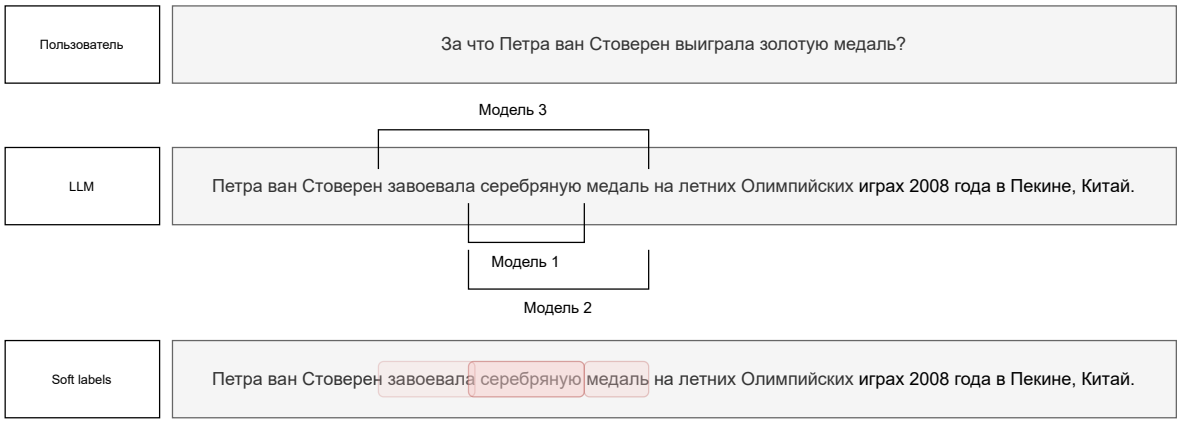

Рис. 4: Пример выделения

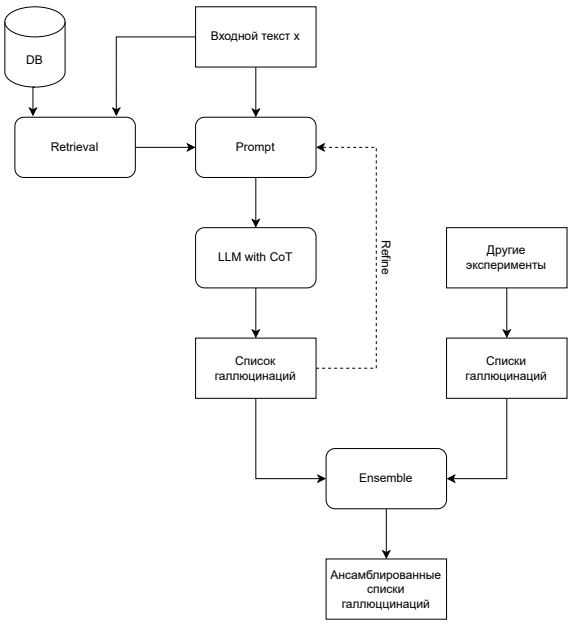

Рис. 5: Конвеер обработки текста для обнаружения галлюцинаций

3.3 Методика оценивания

Для оценки способности модели выделять галлюцинации предлагается сравнивать с эталонным решением. Для оценки близости предложенного набора галлюцинаций с эталонным используются коэффициентом Жаккара (intersection over union, IoU) и корреляция пирсона (corr) — две различные метрики, используемые в статистике и анализе данных для различных целей.

Корреляция Пирсона — это мера линейной зависимости между двумя переменными. Она вычисляется как ковариация двух переменных, деленная на произведение их стандартных отклонений. Коэффициент корреляции Пирсона принимает значение в диапазоне от -1 до 1.

- 1 означает идеальную положительную линейную связь: когда одна переменная увеличивается, другая также увеличивается линейно.
- 0 указывает на отсутствие линейной связи между переменными.
- -1 означает идеальную отрицательную линейную связь: когда одна переменная увеличивается, другая уменьшается линейно.

Формула для вычисления коэффициента корреляции Пирсона ρ между двумя переменными X и Y выглядит следующим образом:

$$\rho(P, Q) = \frac{\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2}\sqrt{\sum_{i=1}^{n}(q_i - \bar{q})^2}}$$

где:

$P = (p_i)_{i=1}^{n}$ — вероятности, что i-й токен принадлежит галлюцинации

$Q = (q_i)_{i=1}^{n}$ — вероятности галлюцинации на выходе модели детекции

Корреляция Пирсона используется для оценки модели с учётом уверенности/неуверности асессоров.

Коэффициент Жаккара, также известный как Intersection over Union (IoU), является мерой схожести между двумя множествами. Особенно часто применяется в области компьютерного зрения для оценки качества распознавания объектов, где нужно определить, насколько предсказанные и истинные области пересекаются.

Формула для коэффициента Жаккара выглядит следующим образом:

$$\text{IoU}(P, Q) = \frac{|P \cap Q|}{|P \cup Q|}$$

$|P \cap Q| = \sum_{i=1}^{n}[p_i > 0.5 \text{ и } q_i > 0.5]$

$|P \cup Q| = \sum_{i=1}^{n}[p_i > 0.5 \text{ или } q_i > 0.5]$

IoU принимает значения в диапазоне от 0 до 1: 0 означает, что множества не пересекаются, 1 указывает на полное совпадение множеств.

В контексте задач, связанных с машинным обучением и анализом изображений, использование IoU позволяет количественно оценить производительность алгоритмов сегментации и детекции объектов.

IoU оценивает конечные предсказания модели, сравнивая фрагменты с финальным решением асессоров.

4 Данные

4.1 SemEval-2025, Задача 3 — Mu-SHROOM

В рамках конкурса SemEval-2025 был предоставлен набор данных, размеченный для задачи обнаружения фактологических галлюцинаций. Данные разделены на две части: для проверки и для тестирования.

Часть для проверки состоит из пяти подвыборок на десяти разных языках, в каждой из которых по 50 объектов. Часть для тестирования включает в себя четыре подвыборки на 14 языках, в каждой из которых по 150 объектов.

Каждый объект в наборе данных представляет собой словарь с полями, описывающими входные данные для модели, её ответ, вероятность галлюцинаций для токенов и другую вспомогательную информацию. Ниже приведён пример одного из объектов:

Листинг 1: Пример объекта выборки

```
{
  "lang": "EN",
  "model_input": "In which city was David Sandberg born?",
  "model_output_text": "David Sandburg was born in Stockholm, Sweden.
     ",
  "model_id": "tiiuae/falcon-7b-instruct",
  "soft_labels": [
    {"start": 27, "prob": 0.909, "end": 36},
    {"start": 36, "prob": 0.091, "end": 44}
  ],
  "hard_labels": [[27, 36]],
  "model_output_tokens": [
    "David", " Sand", "burg", " was", " born", " in",
    " Stockholm", ",", " Sweden", ".", "<|endoftext|>"
  ],
  "model_output_logits": [
    -5.99, -14.99, -11.57, -12.78, -7.70,
    -9.53, -4.74, -8.40, -9.93, -13.37, -8.34
  ]
}
```

В поле model_input содержится исходный запрос, а в model_output_text — текст,

сгенерированный моделью. Модель, которая создала этот текст, имеет идентификатор model_id.

Массив soft_labels показывает вероятность того, что соответствующий токен является галлюцинацией. hard_labels — это бинарная аннотация, которая указывает на позицию токенов, однозначно помеченных как галлюцинации.

Для каждого токена доступны логиты model_output_logits и текстовое представление model_output_tokens.

Каждый объект был размечен 10 разметчиками, после чего их разметки были усреднены, чтобы получить soft_labels.

Особенность этого набора данных в том, что он содержит реальные галлюцинации языковых моделей, а не искусственно созданные путём запроса в промпте.

Для ответа на каждый вопрос авторы запускали несколько моделей и конфигураций, что повышало вероятность галлюцинаций.

Были намеренно использованы не самые большие языковые модели, так как вероятность их галлюцинаций выше.

Набор данных охватывает широкий спектр тем: биология, история, спорт, технологии, география, литература и другие.

Такая структура позволяет использовать набор данных как для обучения моделей в формате sequence labeling, так и для детального анализа ошибок и уверенности модели в своих предсказаниях.

Листинг 2: Инструкция для разметчиков

```
Carefully read the answer text:
- Highlight each span of text in the answer text that is not
   supported by the provided context (i.e., contains an
   overgeneration or hallucination).
- Your annotations should include only the minimum number of
   characters in the text that should be edited/deleted to provide a
   correct answer (in the case of Chinese, these will be "character
   components").
- You are encouraged to annotate conservatively and focus on content
   words rather than function words. This is not a strict guideline,
   and you should rely on your best judgments.
- If the answer text does not contain a hallucination, write "NO
   HALLUCINATION" in the comment box.
```

```
- If you are unsure about how to annotate an example, write "UNSURE"
  in the comment box. Please only use this option as a last resort.
- Ensure that you double-check your annotations. From the "See
  previous annotations" link, you can edit or delete previous
  annotations.
```

4.2 Набор данных и база данных для RAG

Для повышения точности обнаружения галлюцинаций наша система использует конвейер RAG, который включает внешние источники знаний. Мы используем набор данных Wikipedia, только английскую подвыборку с 6,41 млн статей, в качестве основного фактологического источника. Также мы очищаем все статьи, удаляя ссылки и повторяющиеся символы перевода строки. Википедия служит источником для проверки или валидации фактов, использованных в ответе модели. Поиск информации в статьях может помочь выявить неточности или галлюцинации, направив модель к исправлениям или уточнениям баз данных.

Для эффективного поиска мы используем Qdrant[2] как векторную базу данных, которая обеспечивает быстрый и масштабируемый поиск по сходству. Мы используем модель встраивания Multilingual-E5-Large (31) для генерации плотных векторных представлений текста. Чтобы оптимизировать производительность поиска и эффективность хранения, мы встраиваем только первые 512 символов каждой статьи Wikipedia. Сходство между запросами и сохраненными вложениями вычисляется с использованием косинусного расстояния и HNSW в качестве алгоритма поиска.

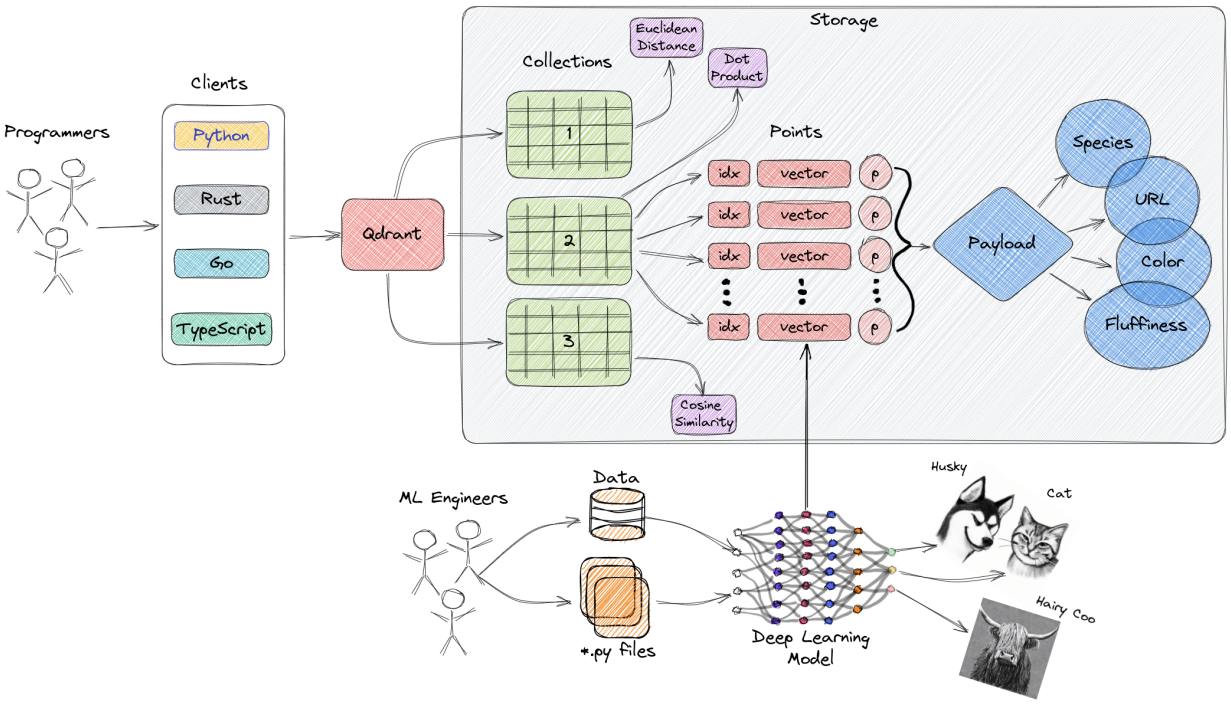

Рис. 6: Общий обзор архитектуры Qdrant[3]

Использование документов из Википедии в промптах языковых моделей может значительно улучшить качество и достоверность генерируемых ответов. Это достигается путем

[2] https://qdrant.tech/

интеграции внешних знаний и предоставления модели актуальных и проверенных данных. Включение выдержек из релевантных статей Википедии в промпт может обогатить контекст, предоставляемый модели. Это особенно полезно для сложных или специализированных вопросов, где точность фактов имеет критическое значение. Например, при запросе информации об историческом событии, тексте из соответствующей статьи можно использовать, чтобы модель опиралась на факты, запрашиваемые пользователем.

5 Эксперименты

Все инструкции для языковых моделей представлены в приложении.

5.1 Без CoT

В качестве модели использовалось GPT-4o от OpenAI В результате экспериментов получили что трехэтапная конвейерная обработка (RAG + самоуточнение + ансамбль) с использованием ансамблирования значительно улучшает выявление галлюцинаций, обеспечивая баланс между проверкой на основе извлечения, самокоррекцией и устойчивостью ансамбля для повышения общего качества. Данный подход достигает 65,09% IoU и 62,94% Corr.

5.2 CoT

В качестве модели для которой можно использовать CoT используется o1-mini от OpenAI и R1 от DeepSeek. Доступ осуществляется через API OpenAI и DeepSeek.

5.3 Результаты

Для более детального анализа влияния техники цепи размышлений (Chain of Thought, CoT) на обнаружение галлюцинаций, мы провели дополнительные эксперименты с использованием моделей o1 от OpenAI и R1 от DeepSeek. В этих экспериментах мы сравнивали результаты с использованием CoT, самоуточнения (self-refine) и ансамблирования.

Перед тем как пробовать лучшие модели, подбирался промпт на более простой моделе (QWEN 70b). Результаты представлены в таблице 1. Финальные инструкции зафиксированы в приложении. QWEN не обучен для CoT, поэтому в инструкции дополнительно просится указывать свои размышления.

Листинг 3: Пример размышлений для QWEN

```
{
    "explanation": 'My task is to check information about arthropods
       having antennaes. According to the relevant document: "Except
       for the chelicerates and proturans, which have none, all non-
       crustacean arthropods have a single pair of antennae.", there
       are arthropods that have no antennaes, so the correct answer
```

```
        is "No", the first part of answer is hallucination. There is
        no mention about visability of antennaes, so this part of
        answer is also hallucination. The minimal way to edit answer
        is "No, all non-crustacean arthropods have antennas. However,
        not all of them have the same function.".',
    "hallucinations": ['Yes', 'arachnids', 'visible', 'naked eye']
}
```

Результаты указывают на то что для модели QWEN техники RAG, refine, ensemble улучшают качество детекции.

Prompt	IoU	Corr
no RAG + CoT	0.36	0.33
RAG + CoT	0.5	0.49
RAG + CoT + refine	0.53	0.5
RAG + CoT + refine + ensemble	0.56	0.52

Таблица 1: QWEN 70b

В результате экспериментов с более тяжёлыми моделями получилось установить что техника самоуточнения не помогает улучшить качество для моделей с цепью размышлений. Результаты для o1-mini и R1 показаны в таблицах ниже (Таблицы 2, 3).

Prompt	IoU	Corr
RAG + CoT	0.59	0.54
RAG + CoT + refine	0.55	0.53
RAG + CoT + refine + ensemble	0.67	0.65

Таблица 2: o1-mini

Результаты для модели o1:

- RAG + CoT: IoU: 0.59, Corr: 0.54, Этот подход показал улучшение по сравнению с базовым RAG, но результаты оставались ниже ожидаемых.
- RAG + CoT + самоуточнение (refine): IoU: 0.53, Corr: 0.51, Добавление самоуточнения не привело к улучшению результатов, что может указывать на то, что модель уже достигла оптимального уровня точности с использованием CoT.

Prompt	IoU	Corr
RAG + CoT	0.61	0.57
RAG + CoT + refine	0.56	0.54
RAG + CoT + refine + ensemble	0.63	0.59

Таблица 3: R1

- RAG + CoT + самоуточнение + ансамблирование: IoU: 0.62, Corr: 0.60, Ансамблирование нескольких моделей с CoT и самоуточнением позволило достичь наилучших результатов, что подтверждает гипотезу о том, что комбинирование методов может значительно повысить точность обнаружения галлюцинаций.

Результаты для модели R1:

- RAG + CoT: IoU: 0.61, Corr: 0.57, Модель R1 показала немного лучшие результаты по сравнению с o1 при использовании CoT.
- RAG + CoT + самоуточнение (refine): IoU: 0.56, Corr: 0.54, Как и в случае с o1, самоуточнение не улучшило результаты, что может быть связано с особенностями архитектуры модели.
- RAG + CoT + самоуточнение + ансамблирование: IoU: 0.63, Corr: 0.59, Ансамблирование снова показало наилучшие результаты, подтверждая его эффективность в сочетании с CoT.
- CoT значительно улучшает качество обнаружения галлюцинаций по сравнению с базовыми подходами.
- Самоуточнение не всегда приводит к улучшению результатов, особенно в сочетании с CoT.
- ансамблирование нескольких моделей с использованием CoT и самоуточнения позволяет достичь наилучших результатов.

6 Вывод

В данной работе была проведена оценка различных методов обнаружения галлюцинаций в больших языковых моделях (LLM). Основные выводы можно сформулировать следующим образом:

- Проблема галлюцинаций:

  Галлюцинации остаются критической проблемой для LLM, особенно в контексте приложений, требующих высокой точности, таких как медицинская или юридическая документация. Обнаружение и минимизация галлюцинаций являются важными задачами для повышения надежности моделей.

- Эффективность методов:

  - RAG (Retrieval-Augmented Generation) значительно улучшает качество обнаружения галлюцинаций за счет использования внешних источников знаний.

  - Chain of Thought (CoT) позволяет моделям более эффективно решать сложные задачи, разбивая их на промежуточные шаги, что улучшает точность обнаружения галлюцинаций.

  - Самоуточнение (self-refine) не всегда приводит к улучшению результатов, особенно в сочетании с CoT.

  - Ансамблирование нескольких моделей позволяет компенсировать ошибки отдельных моделей и повысить общую точность.

- Результаты экспериментов:

  - Комбинация методов (RAG + CoT + ансамблирование) показала наилучшие результаты, достигнув 63% IoU и 59% Corr для модели R1.

  - Самоуточнение не всегда улучшает результаты, что может быть связано с особенностями архитектуры моделей.

- Перспективы:

  - Необходимо продолжать исследования в области интеграции новых методов, таких как CoT и ансамблирование, для повышения устойчивости LLM к галлюцинациям.

  - Разработка более качественных наборов данных для обучения и тестирования моделей на обнаружение галлюцинаций.

  - Изучение влияния различных архитектур моделей на эффективность методов обнаружения галлюцинаций.

В заключение, предложенные методы и подходы позволяют значительно улучшить обнаружение галлюцинаций в LLM, что делает их более надежными и пригодными для использования в реальных приложениях.

В дальнейшем планируется расширить тестируемый корпус, а также провести сравнение предложенного подхода с альтернативными языковыми моделями и методами промптизации.

Представляет интерес исследование автоматических метрик для объективной оценки качества выявления галлюцинаций и привлечение экспертов для ручной валидации выделяемых фрагментов.

Также возможно внедрение предложенного метода в задачи автоматического редактирования и мониторинга качества текстовой генерации.

Дополнительным направлением развития может стать адаптация методик для анализа галлюцинаций в текстах на других языках.

Дальнейшим развитием может стать слабо контролируемая тонкая настройка, которая показала хорошие результаты в аналогичном соревновании SemEval2024.

Список литературы

[1] Asgari, E., Montaña-Brown, N., Dubois, M., Khalil, S., Balloch, J., Yeung, J.A., Pimenta, D.: A framework to assess clinical safety and hallucination rates of llms for medical text summarisation. npj Digital Medicine **8**(1), 1–15 (2025)

[2] Buszydlik, A., Dobiczek, K., Okoń, M.T., Skublicki, K., Lippmann, P., Yang, J.: Red teaming for large language models at scale: Tackling hallucinations on mathematics tasks. arXiv preprint arXiv:2401.00290 (2023)

[3] Cahyawijaya, S., Lovenia, H., Fung, P.: Llms are few-shot in-context low-resource language learners. arXiv preprint arXiv:2403.16512 (2024)

[4] Dahl, M., Magesh, V., Suzgun, M., Ho, D.E.: Large legal fictions: Profiling legal hallucinations in large language models. Journal of Legal Analysis **16**(1), 64–93 (2024)

[5] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)

[6] Grigoriadou, N., Lymperaiou, M., Filandrianos, G., Stamou, G.: Ails-ntua at semeval-2024 task 6: Efficient model tuning for hallucination detection and analysis. arXiv preprint arXiv:2404.01210 (2024)

[7] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)

[8] Hassan, M.: Measuring the impact of hallucinations on human reliance in llm applications. Journal of Robotic Process Automation, AI Integration, and Workflow Optimization **10**(1), 10–20 (2025)

[9] He, P., Liu, X., Gao, J., Chen, W.: Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654 (2020)

[10] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems **43**(2), 1–55 (2025)

[11] Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al.: Openai o1 system card. arXiv preprint arXiv:2412.16720 (2024)

[12] Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. vol. 1, p. 2. Minneapolis, Minnesota (2019)

[13] Kim, Y., Jeong, H., Chen, S., Li, S.S., Lu, M., Alhamoud, K., Mun, J., Grau, C., Jung, M., Gameiro, R., et al.: Medical hallucinations in foundation models and their impact on healthcare. arXiv preprint arXiv:2503.05777 (2025)

[14] Lee, U., Jung, H., Jeon, Y., Sohn, Y., Hwang, W., Moon, J., Kim, H.: Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education. Education and Information Technologies **29**(9), 11483–11515 (2024)

[15] Li, H., Chi, H., Liu, M., Yang, W.: Look within, why llms hallucinate: A causal perspective. arXiv preprint arXiv:2407.10153 (2024)

[16] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)

[17] Liu, T., Zhang, Y., Brockett, C., Mao, Y., Sui, Z., Chen, W., Dolan, B.: A token-level reference-free hallucination detection benchmark for free-form text generation. arXiv preprint arXiv:2104.08704 (2021)

[18] Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., Ge, B.: Summary of chatgpt-related research and perspective towards the future of large language models. Meta-Radiology **1**(2), 100017 (Sep 2023). doi:10.1016/j.metrad.2023.100017, `http://dx.doi.org/10.1016/j.metrad.2023.100017`

[19] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

[20] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al.: Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems **36**, 46534–46594 (2023)

[21] Mickus, T., Zosa, E., Vázquez, R., Vahtola, T., Tiedemann, J., Segonne, V., Raganato, A., Apidianaki, M.: Semeval-2024 task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In: Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024). pp. 1979–1993 (2024)

[22] More, R., Bradbury, J.S.: An analysis of llm fine-tuning and few-shot learning for flaky test detection and classification. arXiv preprint arXiv:2502.02715 (2025)

[23] Nachane, S.S., Gramopadhye, O., Chanda, P., Ramakrishnan, G., Jadhav, K.S., Nandwani, Y., Raghu, D., Joshi, S.: Few shot chain-of-thought driven reasoning to prompt llms for open ended medical question answering. arXiv preprint arXiv:2403.04890 (2024)

[24] Procko, T.: Graph retrieval-augmented generation for large language models: A survey. Available at SSRN (2024)

[25] Rawte, V., Sheth, A., Das, A.: A survey of hallucination in large foundation models. arXiv preprint arXiv:2309.05922 (2023)

[26] Simhi, A., Herzig, J., Szpektor, I., Belinkov, Y.: Distinguishing ignorance from error in llm hallucinations. arXiv preprint arXiv:2410.22071 (2024)

[27] Sukhbaatar, S., Golovneva, O., Sharma, V., Xu, H., Lin, X.V., Rozière, B., Kahn, J., Li, D., Yih, W.t., Weston, J., et al.: Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. arXiv preprint arXiv:2403.07816 (2024)

[28] Tao, L., Li, Y.: Your weak llm is secretly a strong teacher for alignment. arXiv preprint arXiv:2409.08813 (2024)

[29] Tonmoy, S., Zaman, S., Jain, V., Rani, A., Rawte, V., Chadha, A., Das, A.: A comprehensive survey of hallucination mitigation techniques in large language models. arXiv preprint arXiv:2401.01313 (2024)

[30] Vázquez, R., Mickus, T., Zosa, E., Vahtola, T., Tiedemann, J., Sinha, A., Segonne, V., Sánchez-Vega, F., Raganato, A., Libovický, J., Karlgren, J., Ji, S., Helcl, J., Guillou, L., de Gibert, O., Bengoetxea, J., Attieh, J., Apidianaki, M.: SemEval-2025 Task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes (2025), https://helsinki-nlp.github.io/shroom/

[31] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., Wei, F.: Multilingual e5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672 (2024)

[32] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)

[33] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

[34] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems **36**, 11809–11822 (2023)

7 Приложение

Листинг 4: Промпт для COT с GPT

```
[
{{'role':'system', 'content':
"""You are a fact-checking assistant for analysing model
   hallucinations.
Your task is to identify fragments in model_output that are
   hallucinations - parts of the text that are factually incorrect or
    made up by model or inconsistent with model_input.
Dates, names, persons, places, proper nouns (words with capital
   letter), years, numbers in model_output are often hallucinations
   and factually incorrect. Check them carefully.
Detect only hallucination fragments, without neighbour words, that
   are not hallucinations at their own. Don't classify typos as
   hallucinations.
You get a user query in model_input and hallucinated answer in
   model_output. You also get a reliable relevant document from
   Wikipedia, pay attention to this document while checking facts in
   hallucinated model_output. This document can be long. It ends with
    <End of Relevant document 1>.
Write answer in JSON with the next structure:
{{
    "hallucinations": ["h1", "h2"]
}}
where h1 and h2 are hallucination fragments from model_output. Write
   in answer only JSON structure without other comments.

Here is an example of correct dialogue:



Relevant document 1:
Title:
Antenna (biology)
Content:
```

Antennae (: antenna), sometimes referred to as "feelers", are paired appendages used for sensing in arthropods.

Antennae are connected to the first one or two segments of the arthropod head. They vary widely in form but are always made of one or more jointed segments. While they are typically sensory organs, the exact nature of what they sense and how they sense it is not the same in all groups. Functions may variously include sensing touch, air motion, heat, vibration (sound), and especially smell or taste. Antennae are sometimes modified for other purposes, such as mating, brooding, swimming, and even anchoring the arthropod to a substrate. Larval arthropods have antennae that differ from those of the adult. Many crustaceans, for example, have free-swimming larvae that use their antennae for swimming. Antennae can also locate other group members if the insect lives in a group, like the ant.

The common ancestor of all arthropods likely had one pair of uniramous (unbranched) antenna-like structures, followed by one or more pairs of biramous (having two major branches) leg-like structures, as seen in some modern crustaceans and fossil trilobites. Except for the chelicerates and proturans, which have none, all non-crustacean arthropods have a single pair of antennae.

Crustaceans

Crustaceans bear two pairs of antennae. The pair attached to the first segment of the head are called primary antennae or antennules. This pair is generally uniramous, but is biramous in crabs and lobsters and remipedes. The pair attached to the second segment are called secondary antennae or simply antennae. The second antennae are plesiomorphically biramous, but many species later evolved uniramous pairs. The second antennae may be significantly reduced (e.g. remipedes) or apparently absent (e.g. barnacles).

The subdivisions of crustacean antennae have many names, including flagellomeres (a shared term with insects), annuli, articles, and segments. The terminal ends of crustacean antennae have two major categorizations: segmented and flagellate. An antenna is considered segmented if each of the annuli is separate from those around it and has individual muscle attachments. Flagellate antennae, on the other hand, have muscle attachments only around the base, acting as a hinge for the flagelluma flexible string of annuli with no muscle attachment.

There are several notable non-sensory uses of antennae in crustaceans. Many crustaceans have a mobile larval stage called a nauplius, which is characterized by its use of antennae for swimming. Barnacles, a highly modified crustacean, use their antennae to attach to rocks and other surfaces. The second antennae in the burrowing Hippoidea and Corystidae have setae that interlock to form a tube or "snorkel" which funnels filtered water over the gills.

Insects

Insects evolved from prehistoric crustaceans, and they have secondary antennae like crustaceans, but not primary antennae. Antennae are the primary olfactory sensors of insects and are accordingly well-equipped with a wide variety of sensilla (singular: sensillum). Paired, mobile, and segmented, they are located between the eyes on the forehead. Embryologically, they represent the appendages of the second head segment.

All insects have antennae, however they may be greatly reduced in the larval forms. Amongst the non-insect classes of the Hexapoda, both Collembola and Diplura have antenna, but Protura do not.

Antennal fibrillae play an important role in Culex pipiens mating practices. The erection of these fibrillae is considered to be the first stage in reproduction. These fibrillae serve different

functions across the sexes. As antennal fibrillae are used by female C. pipiens to locate hosts to feed on, male C. pipiens utilize them to locate female mates.

Structure

The three basic segments of the typical insect antenna are the scape or scapus (base), the pedicel or pedicellus (stem), and finally the flagellum, which often comprises many units known as flagellomeres. The pedicel (the second segment) contains the Johnston's organ which is a collection of sensory cells.

The scape is mounted in a socket in a more or less ring-shaped sclerotised region called the torulus, often a raised portion of the insect's head capsule. The socket is closed off by the membrane into which the base of the scape is set. However, the antenna does not hang free on the membrane, but pivots on a rigidly sprung projection from the rim of the torulus. That projection on which the antenna pivots is called the antennifer. The whole structure enables the insect to move the antenna as a whole by applying internal muscles connected to the scape. The pedicel is flexibly connected to the distal end of the scape and its movements in turn can be controlled by muscular connections between the scape and pedicel. The number of flagellomeres can vary greatly between insect species, and often is of diagnostic importance.

True flagellomeres are connected by membranous linkage that permits movement, though the flagellum of "true" insects does not have any intrinsic muscles. Some other Arthropoda do however have intrinsic muscles throughout the flagellum. Such groups include the Symphyla, Collembola and Diplura. In many true insects, especially the more primitive groups such as Thysanura and Blattodea, the flagellum partly or entirely consists of a flexibly connected string of small ring-shaped annuli. The annuli are not true flagellomeres, and in a given insect species the number of

annuli generally is not as consistent as the number of flagellomeres in most species.

In many beetles and in the chalcidoid wasps, the apical flagellomeres form a club shape, called the clava. The collective term for the segments between the club and the antennal base is the funicle; traditionally in describing beetle anatomy, the term "funicle" refers to the segments between the club and the scape. However, traditionally in working on wasps the funicle is taken to comprise the segments between the club and the pedicel.

Quite commonly the funicle beyond the pedicel is quite complex in Endopterygota such as beetles, moths and Hymenoptera, and one common adaptation is the ability to fold the antenna in the middle, at the joint between the pedicel and the flagellum. This gives an effect like a "knee bend", and such an antenna is said to be geniculate. Geniculate antennae are common in the Coleoptera and Hymenoptera. They are important for insects like ants that follow scent trails, for bees and wasps that need to "sniff" the flowers that they visit, and for beetles such as Scarabaeidae and Curculionidae that need to fold their antennae away when they self-protectively fold up all their limbs in defensive attitudes.

Because the funicle is without intrinsic muscles, it generally must move as a unit, in spite of being articulated. However, some funicles are complex and very mobile. For example, the Scarabaeidae have lamellate antennae that can be folded tightly for safety or spread openly for detecting odours or pheromones. The insect manages such actions by changes in blood pressure, by which it exploits elasticity in walls and membranes in the funicles, which are in effect erectile.

In the groups with more uniform antennae (for example: millipedes), all segments are called antennomeres. Some groups have a simple or variously modified apical or subapical bristle called an arista (this may be especially well-developed in various Diptera).

Functions

Olfactory receptors on the antennae bind to free-floating molecules, such as water vapour, and odours including pheromones. The neurons that possess these receptors signal this binding by sending action potentials down their axons to the antennal lobe in the brain. From there, neurons in the antennal lobes connect to mushroom bodies that identify the odour. The sum of the electrical potentials of the antennae to a given odour can be measured using an electroantennogram.

In the monarch butterfly, antennae are necessary for proper time-compensated solar compass orientation during migration. Antennal clocks exist in monarchs, and they are likely to provide the primary timing mechanism for sun compass orientation.

In the African cotton leafworm, antennae have an important function in signaling courtship. Specifically, antennae are required for males to answer the female mating call. Although females do not require antennae for mating, a mating that resulted from a female without antennae was abnormal.

In the diamondback moth, antennae serve to gather information about a host plant's taste and odor. After the desired taste and odor has been identified, the female moth will deposit her eggs onto the plant. Giant swallowtail butterflies also rely on antenna sensitivity to volatile compounds to identify host plants. It was found that females are actually more responsive with their antenna sensing, most likely because they are responsible for oviposition on the correct plant.

In the crepuscular hawk moth (Manduca sexta), antennae aid in flight stabilization. Similar to halteres in Dipteran insects, the antennae transmit coriolis forces through the Johnston's organ that can then be used for corrective behavior. A series of low-

```
   light, flight stability studies in which moths with flagellae
   amputated near the pedicel showed significantly decreased flight
   stability over those with intact antennae. To determine whether
   there may be other antennal sensory inputs, a second group of
   moths had their antennae amputated and then re-attached, before
   being tested in the same stability study. These moths showed
   slightly decreased performance from intact moths, indicating there
    are possibly other sensory inputs used in flight stabilization.
   Re-amputation of the antennae caused a drastic decrease in flight
   stability to match that of the first amputated group.
<End of Relevant document 1>


model input: Do all arthropods have antennae?


model output: Yes, all arachnids have antennas. However, not all of
   them are visible to the naked eye.


Your answer:
{{
    "hallucinations": ["Yes", "arachnids", "visible", "naked eye"]
}}"""}},

{{'role':'user', 'content':
"""Relevant document 1:
{doc_1}
<End of Relevant document 1>


model_input: {model_input}


model_output: {model_output_text}
```

```
Your answer:"""}}
]
```

Листинг 5: Промпт для refine

```
[
{{'role':'system', 'content': """You are a fact-checking assistant
   for analysing model hallucinations. Your task is to identify
   fragments in model output that are hallucinations - parts of the
   text that are factually incorrect or made up by model or
   inconsistent with model input. You get a user query in model input
    and hallucinated answer in model output. You get a reliable
   relevant document from Wikipedia, pay attention to this document
   while checking facts in hallucinated model output. You will also
   get answer from another model, this answer may be incorrect, take
   attention to this answer, try to fix it. Detect only hallucination
    fragments, without neighbour common, linking words. Write answer
   in JSON with the next structure:
{{
  "hallucinations": ["h1", "h2"]
}}
where h1 and h2 are hallucination fragments from model output. Write
   in answer only JSON structure without other comments. Try to make
   hallucination fragments shorter if you can. If there is year in
   hallucination fragment, write only year number.

Here is an example of correct dialogue:



Relevant document 1:
Title:
Antenna (biology)
Content:
Antennae (: antenna), sometimes referred to as "feelers", are paired
   appendages used for sensing in arthropods.

Antennae are connected to the first one or two segments of the
   arthropod head. They vary widely in form but are always made of
   one or more jointed segments. While they are typically sensory
```

organs, the exact nature of what they sense and how they sense it is not the same in all groups. Functions may variously include sensing touch, air motion, heat, vibration (sound), and especially smell or taste. Antennae are sometimes modified for other purposes, such as mating, brooding, swimming, and even anchoring the arthropod to a substrate. Larval arthropods have antennae that differ from those of the adult. Many crustaceans, for example, have free-swimming larvae that use their antennae for swimming. Antennae can also locate other group members if the insect lives in a group, like the ant.

The common ancestor of all arthropods likely had one pair of uniramous (unbranched) antenna-like structures, followed by one or more pairs of biramous (having two major branches) leg-like structures, as seen in some modern crustaceans and fossil trilobites. Except for the chelicerates and proturans, which have none, all non-crustacean arthropods have a single pair of antennae.

Crustaceans

Crustaceans bear two pairs of antennae. The pair attached to the first segment of the head are called primary antennae or antennules. This pair is generally uniramous, but is biramous in crabs and lobsters and remipedes. The pair attached to the second segment are called secondary antennae or simply antennae. The second antennae are plesiomorphically biramous, but many species later evolved uniramous pairs. The second antennae may be significantly reduced (e.g. remipedes) or apparently absent (e.g. barnacles).

The subdivisions of crustacean antennae have many names, including flagellomeres (a shared term with insects), annuli, articles, and segments. The terminal ends of crustacean antennae have two major categorizations: segmented and flagellate. An antenna is considered segmented if each of the annuli is separate from those around it and has individual muscle attachments. Flagellate

antennae, on the other hand, have muscle attachments only around the base, acting as a hinge for the flagelluma flexible string of annuli with no muscle attachment.

There are several notable non-sensory uses of antennae in crustaceans. Many crustaceans have a mobile larval stage called a nauplius, which is characterized by its use of antennae for swimming. Barnacles, a highly modified crustacean, use their antennae to attach to rocks and other surfaces. The second antennae in the burrowing Hippoidea and Corystidae have setae that interlock to form a tube or "snorkel" which funnels filtered water over the gills.

Insects

Insects evolved from prehistoric crustaceans, and they have secondary antennae like crustaceans, but not primary antennae. Antennae are the primary olfactory sensors of insects and are accordingly well-equipped with a wide variety of sensilla (singular: sensillum). Paired, mobile, and segmented, they are located between the eyes on the forehead. Embryologically, they represent the appendages of the second head segment.

All insects have antennae, however they may be greatly reduced in the larval forms. Amongst the non-insect classes of the Hexapoda, both Collembola and Diplura have antenna, but Protura do not.

Antennal fibrillae play an important role in Culex pipiens mating practices. The erection of these fibrillae is considered to be the first stage in reproduction. These fibrillae serve different functions across the sexes. As antennal fibrillae are used by female C. pipiens to locate hosts to feed on, male C. pipiens utilize them to locate female mates.

Structure

The three basic segments of the typical insect antenna are the scape or scapus (base), the pedicel or pedicellus (stem), and finally the flagellum, which often comprises many units known as flagellomeres. The pedicel (the second segment) contains the Johnston's organ which is a collection of sensory cells.

The scape is mounted in a socket in a more or less ring-shaped sclerotised region called the torulus, often a raised portion of the insect's head capsule. The socket is closed off by the membrane into which the base of the scape is set. However, the antenna does not hang free on the membrane, but pivots on a rigidly sprung projection from the rim of the torulus. That projection on which the antenna pivots is called the antennifer. The whole structure enables the insect to move the antenna as a whole by applying internal muscles connected to the scape. The pedicel is flexibly connected to the distal end of the scape and its movements in turn can be controlled by muscular connections between the scape and pedicel. The number of flagellomeres can vary greatly between insect species, and often is of diagnostic importance.

True flagellomeres are connected by membranous linkage that permits movement, though the flagellum of "true" insects does not have any intrinsic muscles. Some other Arthropoda do however have intrinsic muscles throughout the flagellum. Such groups include the Symphyla, Collembola and Diplura. In many true insects, especially the more primitive groups such as Thysanura and Blattodea, the flagellum partly or entirely consists of a flexibly connected string of small ring-shaped annuli. The annuli are not true flagellomeres, and in a given insect species the number of annuli generally is not as consistent as the number of flagellomeres in most species.

In many beetles and in the chalcidoid wasps, the apical flagellomeres form a club shape, called the clava. The collective term for the segments between the club and the antennal base is the funicle;

traditionally in describing beetle anatomy, the term "funicle" refers to the segments between the club and the scape. However, traditionally in working on wasps the funicle is taken to comprise the segments between the club and the pedicel.

Quite commonly the funicle beyond the pedicel is quite complex in Endopterygota such as beetles, moths and Hymenoptera, and one common adaptation is the ability to fold the antenna in the middle, at the joint between the pedicel and the flagellum. This gives an effect like a "knee bend", and such an antenna is said to be geniculate. Geniculate antennae are common in the Coleoptera and Hymenoptera. They are important for insects like ants that follow scent trails, for bees and wasps that need to "sniff" the flowers that they visit, and for beetles such as Scarabaeidae and Curculionidae that need to fold their antennae away when they self-protectively fold up all their limbs in defensive attitudes.

Because the funicle is without intrinsic muscles, it generally must move as a unit, in spite of being articulated. However, some funicles are complex and very mobile. For example, the Scarabaeidae have lamellate antennae that can be folded tightly for safety or spread openly for detecting odours or pheromones. The insect manages such actions by changes in blood pressure, by which it exploits elasticity in walls and membranes in the funicles, which are in effect erectile.

In the groups with more uniform antennae (for example: millipedes), all segments are called antennomeres. Some groups have a simple or variously modified apical or subapical bristle called an arista (this may be especially well-developed in various Diptera).

Functions

Olfactory receptors on the antennae bind to free-floating molecules, such as water vapour, and odours including pheromones. The neurons that possess these receptors signal this binding by sending

action potentials down their axons to the antennal lobe in the brain. From there, neurons in the antennal lobes connect to mushroom bodies that identify the odour. The sum of the electrical potentials of the antennae to a given odour can be measured using an electroantennogram.

In the monarch butterfly, antennae are necessary for proper time-compensated solar compass orientation during migration. Antennal clocks exist in monarchs, and they are likely to provide the primary timing mechanism for sun compass orientation.

In the African cotton leafworm, antennae have an important function in signaling courtship. Specifically, antennae are required for males to answer the female mating call. Although females do not require antennae for mating, a mating that resulted from a female without antennae was abnormal.

In the diamondback moth, antennae serve to gather information about a host plant's taste and odor. After the desired taste and odor has been identified, the female moth will deposit her eggs onto the plant. Giant swallowtail butterflies also rely on antenna sensitivity to volatile compounds to identify host plants. It was found that females are actually more responsive with their antenna sensing, most likely because they are responsible for oviposition on the correct plant.

In the crepuscular hawk moth (Manduca sexta), antennae aid in flight stabilization. Similar to halteres in Dipteran insects, the antennae transmit coriolis forces through the Johnston's organ that can then be used for corrective behavior. A series of low-light, flight stability studies in which moths with flagellae amputated near the pedicel showed significantly decreased flight stability over those with intact antennae. To determine whether there may be other antennal sensory inputs, a second group of moths had their antennae amputated and then re-attached, before being tested in the same stability study. These moths showed

```
   slightly decreased performance from intact moths, indicating there
    are possibly other sensory inputs used in flight stabilization.
   Re-amputation of the antennae caused a drastic decrease in flight
   stability to match that of the first amputated group.

References

Arthropod anatomy
<End of Relevant document 1>


model input: Do all arthropods have antennae?


model output: Yes, all arachnids have antennas. However, not all of
   them are visible to the naked eye.

Answer from another model:
{{
  "hallucinations": ["all arachnids", "antennas"]
}}

Your answer:
{{
  "hallucinations": ["Yes", "arachnids", "visible", "naked eye"]
}}"""}},

{{'role':'user', 'content':
"""Relevant document 1:
{doc_1}
<End of Relevant document 1>


model input: {model_input}
```

```
model output: {model_output_text}


Answer from another model:
{response_prev}


Your answer:"""}}
]
```